EXE

AUGUST 1991 VOL 6 ISSUE 3

Eight GUI Libraries for C & C++

The Old War Horses still carrying on.
We look at FORTRAN and COBOL this issue.

After a decade of committee bickering –
The FORTRAN 90 standard.

Designed to be like English!
Putting COBOL through The Third Side.

Getting Help with Microsoft Windows.

Peter Collinson goes into POSIX,
Jules May goes into Hyperspace.

THE TRAINING LEADERS

To C PROGRAMMERS AT ALL LEVELS

The best path for complete C training begins at QA

C has become the dominant language in computer programming and provides access to the software industries major application program interfaces.

C is probably now, or shortly will be, your way forward in program development – and QA can really help you along that path.

Only QA's extensive C curriculum totally supports the training needs of all C programmers, from beginner through to highly experienced professional. Each course can stand alone or can be the entry point to reach any level of skill or knowledge you may set yourself.

In common with all QA courses, our instructors are experienced professionals themselves, our training materials are perfected for efficient learning, and all our own skill and energy is channelled towards helping you get the maximum benefit from your training.

BEGINNERS START HERE

C Primer 2 days

Introduction to HLL concepts via C • An overview of the C programming environment • Developing simple C programs • Data typing, block structure, score concepts • Introducing C functions and modular programming • How to use the C library and preprocessor • Working with larger programs • Introduction to advanced techniques

OTHERS START HERE

C Programming 4 days

C programm structure and style • Data types and variables. Flow of control in C programs • Pointers and address manipulation • Portability and efficiency considerations • Debugging C programs • Working with larger C programs

OR ANYWHERE ELSE

Working with Microsoft C 1 day

Product overview • Compiler memory models • Tools. Programmers workbench • Customising the environment

C Advanced Software Development 4 days

C review • Common problems • ANSI features • Advanced pointers • Standard library • Dynamic memory techniques • Efficiency and optimisation

C Standards 1 day

C standards – aims, goals and benefits • Defining practical and realistic standards • Naming and layout standards • C program organisation standards • Defensive C programming • Portability, maintainability and efficiency

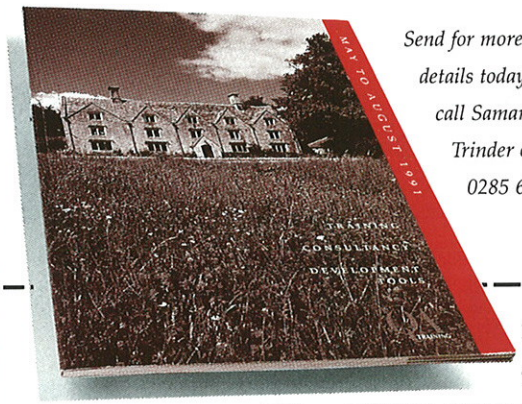
INCLUDING C++

C++ Programming 4 days

C++ today • Declaring and using classes • Operator and function overloading • C++ advanced syntax concepts • Single and multiple inheritance • Performance considerations • Class libraries and application frameworks • OOP: the pitfalls

Object – Oriented Design 3 days

OOP and OOD: the need • OOD: fundamental concepts • Identifying objects and relationships • Object properties • Stages in object-oriented design • C++ design guidelines



Send for more details today or call Samantha Trinder on 0285 655888.

Mail to QA Training Ltd, Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EF. UK.

Tick here for full details:

☐ C ☐ Object-Oriented ☐ OS/2 ☐ Unix

☐ Windows ☐ Networks ☐ Consultancy ☐ Products

NAME _____ TITLE _____

COMPANY _____

ADDRESS _____

TEL. NO. _____

QA TRAINING

C Programming Training For All Levels

CIRCLE NO. 040

Vol 6 Issue 3

.EXE

August 1991

General

.EXE Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH.
Tel: (Advertising/Editorial/Production) 081 994 6477
(Subscriptions) 044282 4501
Facsimile: 081 994 1533 Telex: 8811418 SPACES G
ISSN: 0268-6872

Subscriptions

.EXE Magazine is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (11 issues) in the UK. 'A subscription implies that this journal will be sent to the subscriber until one of the three expires' - AG Macdonell. The magazine is published around the 26th of the month preceding the cover date. There is no January issue. To subscribe or obtain details of overseas/academic rates, please call 044282 4501, or write to The Subscriptions Manager, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We can invoice your company or take ACCESS and VISA cards. In case of a query about your subscription, please call 044282 4501.
Back issues are available at a cost of £3.50 each - call our Chiswick office (081 994 6477) for a list of issues still in print. We also offer a range of magazine binders, disk holders etc - please call for details.

Editorial

Editorial enquiries should be addressed to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We welcome letters, opinions, suggestions and articles from our readers. Please write for a copy of our Contributors' Guide.
Information contained in .EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.
From time to time, we offer to copy the PC program code described in an article onto diskette for our readers. In this case, please send a blank, formatted disk with a self-addressed, prepaid mailer to the editorial address given above. We can copy both 5.25" and 3.5" disks.
The publishers can accept no liability for any consequences of using software distributed in this way.

Advertising

If you are interested in advertising in this magazine, please write to the Display Advertising Manager/Recruitment Sales Manager (as appropriate), .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH, or call 081 994 6477 for details of our advertising rates.

Pronunciation

The name of .EXE Magazine is pronounced to rhyme with 'not sexy magazine'.

Issue theme: FORTRAN and COBOL

FORTRAN 90, THE NEW FORTRAN STANDARD
The new FORTRAN spec has been cooking for 10 years.
John Reid cracks open a hard-boiled standard. *14*

THREE COBOL JEWELS?
Vincent Kilcoyn looks at three MS-DOS COBOL compilers. *21*

WRITING C++ MEMBER FUNCTIONS IN ASSEMBLY LANGUAGE
C++ is not the easiest language to mix with.
Laine Stump has been toiling among the name manglers. *27*

UPPER, MIDDLE OR WORKING CLASS?
This year has seen a proliferation of C and C++ libraries for GUIs.
Paul Kemp and Cliff Saran check out eight of the best. *36*

HE CALLED ME 'SQUIRE'
There are many ways of choosing a software house,
but Darrel Ince's strategy begins with a very fishy observation. *45*

PRESERVING DATA WITH ARK
Stephen Morris investigates a very unusual British spreadsheet program. *52*

HELPING HANDS
It's easier to get help in Windows than it is to give it.
Jeffrey Goldberg explains how to cope with F1. *60*

SOAPBOX
Nick Evans is another who believes that C++
gets more than its fair share of the OOP limelight. *2*

NEWS
Macro: Borland makes a dramatic move,
micro: a new bookshop opens just for us. *4*

LETTERS
Our advertisers take a little flak, and a Simula fan speaks out. *10*

MAYHEM
The inimitable Jules May launches into hyperspace. *68*

THE THIRD SIDE
Christina Wheeler is not quite an apologist for COBOL,
but she does seem to have a soft spot for it. *71*

THE CODE PAGE
Ray Jones sets out his hash tables for fast storage and retrieval. *76*

UNIX REGULAR
POSIX is a portable operating system which isn't UNIX,
but also is, although... Peter Collinson clears things up. *82*

BOOKS
We have one for Windows Power Users, and one for 68k assembly programmers. *88*

CROSSWORD
Another tough puzzle from Eric Deeson. *90*

STOB
Stob has an encounter with some love-story generating software. *96*

O-O environments, not languages

To get the real benefits from OOP, you require more than just a language, argues Nick Evans.

Two technologies are indisputably in fashion this year. One is GUIs. The other is object-oriented programming and development. The two are linked - the reason OOP is becoming so popular is that it is much more productive than C for development under platforms such as Windows, PM and OSF/Motif.

Of all the OOP languages available, the most popular is C++. Due to the increasing popularity of the object-oriented paradigm for development and C++ in particular there is a danger that the two will be seen as identical - that to use object-oriented techniques means to use C++. This is not the case. In fact many of the productivity benefits of using OOP for GUI development are more to do with the environment provided by the language than the syntax.

Not all OOPLs are the same. Let us consider the case of Windows development. There are many OOPLs around now for Windows; for example C++, Actor, Object\1, Enfin and Smalltalk. Let's compare these with the traditional system of Windows development: Microsoft C.

All of these have some form of symbolic debugger. Both C and Glockenspiel C++ use Microsoft CodeView; Actor and Smalltalk have their own debuggers within the environment. The point about CodeView is that it is not integrated with the language used. In Actor or Smalltalk, an error will automatically cause the debugger to be invoked (in the development system). The programmer can then examine any variable, update the database, amend the code and finally resume the application. This makes for a very fast iteration indeed - something that cannot be matched by a system without an integrated debugger.

Incremental compilation is another important point. None of the environments mentioned is interpreted - Actor uses token threading (c/f FORTH), while Smalltalk compiles methods down to an executable. However, in all cases the environment gets away from the traditional code-compile-run cycle, with the tool accepting changes either as they are made and compiling them as they are accepted or when they are next run. Whichever the approach, the system is never compiling more than a page or two of code - so that the time taken to accept changes is rarely more than five seconds.

One important point about environments is that they often perform automatic memory management for the developer. This means that the programmer need never use the `alloc()` and `free()` functions again - the environment reuses space as and when variables are no longer referred to by objects within the system.

Development speed, something of a moot point this - just how much a given object-oriented tool reduces development time depends on which vendor you are speaking to. At my company we have used Actor, Smalltalk, C++ and C on a number of developments and are independent of all of them. This makes us well-placed to compare the development time and resource required to build systems in each of these languages.

It has been our experience that, of the four tools mentioned, Smalltalk/V Windows is the most productive, followed by Actor. C++ is some way behind both of them (although more efficient in terms of execution speed), while C itself is the slowest.

Each of these points boost productivity significantly. The point for us is that development in C++ is significantly faster than C alone, but nowhere near as productive as a full environment. This point is not lost on the vendors of C++ compilers - Borland has recently released a version of C++ surrounded by a sophisticated development environment , Microsoft is promising a class library to support its own C++ compiler shortly after it (eventually) arrives.

The fact is that a language like C++ will always be slower to use for development than a fully supported development environment. In return for this overhead on development, we receive a faster, smaller and more efficient runtime. So the question that has to be asked is: at a time when hardware cost is falling and performance increasing, does it really matter whether a menu pops up in 10 nanoseconds or 60? And the answer must be no - not for the majority of applications.

At a time when developers are switching in droves to use C++, it may be unpopular to point out that there are other, more efficient, ways to build Windows systems. Nevertheless, if you are someone who pays for development, or someone who is responsible for the delivery of Windows apps in short time-scales, then the use of object-oriented environments rather than simply languages may be worth a look. After all, what matters is not how purely a language matches some arbitrary criteria of OOPness - rather how it can help do a particular job.

EXE

Nick Evans spent many years as a consultant in IT before he saw the light and co-founded Information Systems Associates to build systems for people. For the last 3 years ISA (081 840 3422) have been using OOP techniques and languages to deliver Windows and PM systems.

GREY MATTER

Prigg Meadow, Ashburton, Devon TQ13 7DF

Prices do not include VAT or other local taxes, but do include delivery in the UK and Europe. Please check prices at time of order as ads are prepared some weeks before publication. This page only lists some of our products – call us for a complete price list.

ORDER BY PHONE WITH YOUR CREDIT CARD.

TEL: (0363) 53499

FAX: (0364) 53071

PROLOG LANGUAGE

Arity/Prolog Comp	OS/2	£745
ADA Educ.Prolog	MS-DOS	£ 26
ADA FS Prolog	MS-DOS	£ 45
ADA VMI Prolog	MS-DOS	£ 70
ADA VML Prolog	MS-DOS	£125
Arity/Prolog Comp. v6	PC-DOS	£520
Arity/Prolog Int. v6	PC-DOS	£220
Arity Standard	PC-DOS	£ 75
Flex Expert Dev Sys	PC-DOS	£1210
LPA PROLOG Prof v3	PC-DOS	£755
Micro-PROLOG v3.1	MS-DOS	£ 70
PDC Prolog	PC-DOS	£240
Micro-Prolog v3.1	CP/M-86	£ 70
Micro-Prolog v3.1	CP/M-80	£ 70
Prolog-1 v2.2	CP/M-80	£225
LPA Mac Prolog v3	Apple Mac Pls	£450

THE C LANGUAGE

New Topspeed C & C++ V3 now available. New Hi-Tech C 386 PDS generates DOS and 386 code, includes own DOS Extender. Refer also to the C++ section.

C COMPILERS

Microsoft C v6	OS/2&PC-DOS	£222
Topspeed C for OS/2	OS/2	£335
Watcom C v8 Prof	OS/2&DOS	£310
Avocet AVCOM 86C	PC-DOS	£210
Aztec C86 Professional	MS-DOS	£120
Aztec C86 Developer	MS-DOS	£175
Aztec C86 Commercial	MS-DOS	£250
HIGH C v1.61	MS-DOS	£390
HIGH C 386 Global	PL386&MS-DOS	£650
Hi-Tech C	MS-DOS	£165
Hi-Tech C 386 PDS	PC-DOS	£385
Microsoft QuickC v2.5	PC-DOS	£ 50
M'soft QuickC/QuickAsm	PC-DOS	£ 83
MIX C	MS-DOS	£ 20
MIX Power C & lib s'ce	PC-DOS	£ 30
Topspeed C Std	PC-DOS	£110
Topspeed C Prof	PC-DOS	£165
Watcom C v8 Std	MS-DOS	£265
Watcom C/386 Std	PL386&MS-DOS	£585
Watcom C/386 Prof	PL386&MS-DOS	£780
Avocet C Avcom Z80	CP/M-80	£210
Aztec C Personal 1.06D	CP/M-80	£120
Aztec Commercial 1.06D	CP/M-80	£200
Hi-Tech C	Z80 + CP/M-80	£105
Mix C	Z80 + CP/M-80	£ 35
Aztec C68K MPW C	MACINTOSH	£110
Aztec C68K C	MACINTOSH	£ 75
Aztec C68K/AM Developer	AMIGA	£175
Aztec C68K/AM Prof'snl	AMIGA	£120
Aztec C68K/ST Developer	ATARI	£175
Aztec C68K/ST Prof'snl	ATARI	£120
Hi-Tech C 68000	ATARI	£140
Mark Williams C v2	ATARI	£110
Laser C	ATARI	£135
Prospero C	ATARI	£ 80

C CROSS COMPILERS

We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on MS-DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096, 68000 & 68020. Please call for information or advice.

C INTERPRETERS

C-terp V3.0X	PC-DOS	£165
Instant C/16M	PC-DOS	£700
Interactive C	PC-DOS	£195
Introducing C	PC-DOS	£ 85
RUN/C Professional	MS-DOS	£110

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, ISIS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, DEC RT- 11, IBM BEF, ATARI ST, AMSTRAD, MACINTOSH.

Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt.

For more information call us.

PROGRAMMING TOOLS

Ada Compilers
Algol Compilers
Assemblers & Libs
AWK
Basic Compilers
Basic Interpreters
Basic Utilities
Basic Libraries
C Compilers
C Utilities
C Interpreters
C Libraries
C++
Cobol Compilers
Comms.Libraries
Cross Assemblers
Database Libs.
Debuggers
Dis-assemblers
Editors
Engineers Libs.
Expert Systems
Forth
Fortran Compilers
Fortran Libraries
Graphics Libraries
Icon
Linkers/Locaters
Lisp
Logo
Modula-2
Nial Interpreters
Object Oriented
OPS 5
Pascal Compilers
Pascal Libraries
Prolog
Rexx
Risc
Screen Libraries
Smalltalk
Snobol

We stock many items for which there is no space in these advertisements.

DEBUGGING TOOLS

Multiscope for OS/2	OS/2	£315
Multiscope for DOS	PC-DOS	£140
Periscope I/512K	PC-DOS	£475
Periscope II	PC-DOS	£125
Periscope II-X	PC-DOS	£ 95
Quaid Analyser	PC-DOS	£125
Soft-Ice	386&PC-DOS	£340
Soft-Ice + Magic	CV 386&PC-DOS	£440
Turbo Debugger v2	PC-DOS	£ 89

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

Microsoft Basic PDS 7.1	**222.00**
Microsoft C PDS 6.0	**222.00**
Microsoft C/Windows SDK	**364.00**
Microsoft Cobol 4.0	**365.00**
Microsoft Fortran 5.0	**205.00**
Microsoft Macro Assembler 6.0	**63.00**
Microsoft Pascal 4.0	**139.00**
Microsoft QuickBASIC 4.5	**50.00**
Microsoft QuickC 2.51	**50.00**
Microsoft QuickC & Assembler	**83.00**
Microsoft QuickPASCAL 1.0	**50.00**
Microsoft Windows 3.0	**70.00**
Microsoft Windows SDK	**265.00**
Borland C++ 2.0	**206.00**
Turbo C++	**48.00**
Turbo Debugger & Tools 2.0	**89.00**
Turbo Pascal 6.0	**69.00**
Turbo Pascal Professional 6.0	**137.00**
Turbo Pascal for Windows	**103.00**
Object Vision 1.0	**69.00**
Paradox 3.5	**409.00**
Paradox Engine	**223.00**
Paradox MultiPack	**615.00**

WITH FULL TECHNICAL SUPPORT

C LIBRARIES & TOOLS

COMMS LIBRARIES

C Asynch Manager V3s'ce	PC-DOS	£135
Essential Comms(MS&TC)	PC-DOS	£180
Greenleaf Comms(source)	PC-DOS	£195
Multi-Comm (MS)	PC-DOS	£190
Silvercomm C Asynch	PC-DOS	£140

DATABASE/FILEHANDLING

Btrieve for OS/2	OS/2	£360
Btrieve for Windows	MS-DOS	£360
Btrieve v5.1	MS-DOS	£360
XQL	OS/2,PC-DOS	£440
Xtrieve Plus	OS/2,PC-DOS	£360
C/Database T'lchest sce	PC-DOS	£ 39
CBTREE	(source any C)	£135
C-Index/Plus	(source any C)	£175
C-ISAM (L,MS)	MS-DOS	£170
C-tree	(source any C)	£300
C-tree Plus	PC-DOS	£360
Essential Btree (s'ce)	PC-DOS	£110
CQL s'ce	PC-DOS	£270
Faircom Toolbox Prof v2	PC-DOS	£680
Pro-C & Workbench v2	PC-DOS	£585
R-tree	MS-DOS	£140
db-VISTA DBMS Object	PC-DOS	£450
Lattice dBCIII	(L,MS) MS-DOS	£185
Lattice dBCIII+	OS/2&MS-DOS	£365
SftFcus Btree&Isam	(sce any C)	£ 80

GRAPHICS

Essential Graphics v3	PC-DOS	£215
GFX Graphics	PC-DOS	£ 80
GraphiC v5 (MS)	PC-DOS	£245
graphics MENU/META v3	PC-DOS	£125
GSS Kernel system	OS2,PC-DOS	£425
GSS Graph Dev T'kit	OS2,PC-DOS	£425
GX Graphics	PC-DOS	£ 95
HALO v3 (MSC5,TC)	PC-DOS	£215
HALO Professional (C)	PC-DOS	£335
HGraph (MSC)	PC-DOS	£ 80
Ingraf (MSC)	PC-DOS	£185
MetaWINDOW v3.7c	PC-DOS	£145
MetaWINDOW/Plus	PC-DOS	£195
TurboWINDOW/C (Turbo)	PC-DOS	£ 85
QuickWINDOW/C (QC)	PC-DOS	£ 85

SCREEN & WINDOWS

C-Scape +Look&Feel	OS/2&PC-DOS	£390
Vitamin C for OS/2	OS/2&DOS	£240
C View Manager (s'ce)	PC-DOS	£245
Curses/PC w. s'ce	PC-DOS	£235
Entelekon Windows(s'ce)	PC-DOS	£105
Greenleaf Datawindow	PC-DOS	£215
Greenleaf Makeform	PC-DOS	£ 75
Lattice Curses V Lib	PC-DOS	£105
Multi-windows (MS,L)	PC-DOS	£190
Panel Plus II (source)	PC-DOS	£260
Power Screen (MS&TC)	PC-DOS	£ 99
Vermont Views (MSC,TC)	PC-DOS	£345
Vitamin C	(source) PC-DOS	£145

GENERAL FUNCTIONS

C/Utilities T'chest sce	PC-DOS	£ 39
C-Worthy w CW Architect	MS-DOS	£410
C Tools Plus/6.0 (MS)	PC-DOS	£105
CodeRunner	PC-DOS	£ 95
Entelekon Funct(source)	PC-DOS	£105
Essential C Utils v5	PC-DOS	£140
G'leaf Fin'cial Mathlib	PC-DOS	£215
G'leaf Super Functions	PC-DOS	£160
Greenleaf Functions s'ce	PC-DOS	£125
Heap Checker	PC-DOS	£ 30
Lattice C Comp Kit+s'ce	PC-DOS	£160
PDK£1	PC-DOS	£ 70
Power Search	PC-DOS	£ 99
Resident-C (MS,L)(s'ce)	PC-DOS	£135
Silverware C EMM Lib	PC-DOS	£105
The Heap Expander(MS&TC)	PC-DOS	£ 65
Turbo C Tools v2	PC-DOS	£ 95
WKS Library	PC-DOS	£160

MATHEMATICAL LIBRARIES

C Lang Sci Lib v2	ANY C	£160
Mathpak 87 (L,MS)	MS-DOS	£ 80
Science & Eng Tools	MS-DOS	£ 59

TOOLS & SUNDRIES

PC-Lint	OS/2,MS-DOS	£ 80
Clear+ for C	PC-DOS	£140
Databoss (C)	PC-DOS	£320
Lattice Comp Companion	MS-DOS	£ 70

REAL TIME & MULTI-TASKING TOOLS

Concurrent C	(PC/MPX)sceMS-DOS	£ 60
Multi-C (MSC,TC)	PC-DOS	£190
Op.Sys.T'box (MS) s'ce	PC-DOS	£ 69
Timeslicer v6 (MS6)	PC-DOS	£200
Over-C (MS)	PC-DOS	£225

ASSEMBLERS

The new V6 of Microsoft's Macro Assembler is shipping. Call for details.

2500AD 80386 ASM v5	PC-DOS	£230
MS Macro-86 v5.1	OS/2&PC-DOS	£ 76
MS MASM PDS v6	OS/2&PC-DOS	£ 63
PharLap 286 DOS Ext SDK	MS-DOS	£350
PharLap 386 DOS Ext SDK	MS-DOS	£350
PharLap 386 ASM/Linkloc	MS-DOS	£905
PharLap 386/VMM	MS-DOS	£240
Optasm	MS-DOS	£110
Turbo Debugger v2	PC-DOS	£ 89
2500AD Z80 ASM	CP/M-80	£ 70
SLR Z80ASM	CP/M-80	£ 40
SLR Z80ASM-PLUS	CP/M-80	£140
SLR MAC	CP/M-80	£ 40
SLR MAC-PLUS	CP/M-80	£140
SLR 180 (Hitachi)	CP/M-80	£ 40
SLR 180-PLUS (Hitachi)	CP/M-80	£140

Not all assemblers are supplied with a linker. Check before ordering.

BRIEFor C++

Yes, it's a poxy name isn't it, but it looks quite nifty. BRIEFor C++ works with the BRIEF 3.1 editor to provide a class browser for C++ programmers. Features include facilities to report on class hierarchies, member functions and variables, and associated implementation files. BRIEFor C++ for DOS or OS/2 is on special offer at £99 until the end of August and requires a copy of BRIEF 3.1. It is available from Solution Systems on 0763 244141.

New Motif GUI Tool

Protek has announced UK availability of the UIM/X interface generator for the Motif/X Windows environment. Up to now, the product has only been available in the States. In addition to the standard set of GUI tools, UIM/X also contains a C interpreter which allows the behaviour of the user interface to be created, modified, and tested at design time (without recompilation). Prices start at £3,125 for a single-user licence. Protek is on 0895 446000.

Derive V2.0

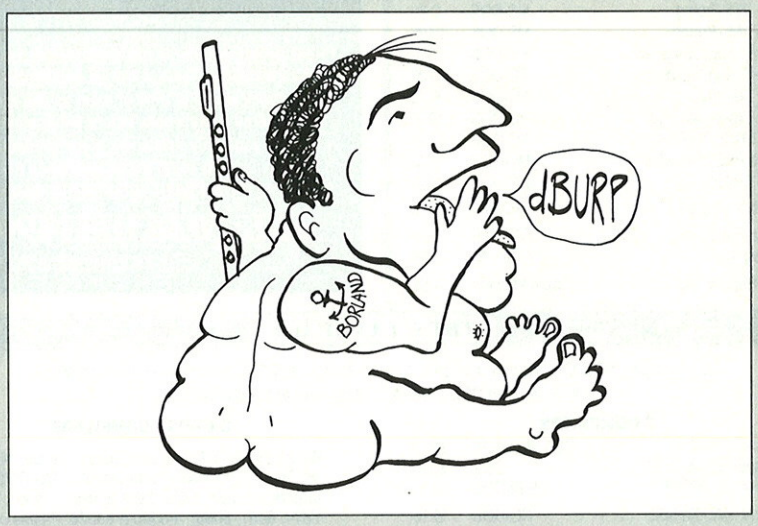
'2000 years of mathematical knowledge on a disk' it says here. Version 2.0 of the maths program contains extras such as recursion, iteration and a programming language. The package takes a non-numeric rule-based approach to problem solving which is claimed to be more accurate than other stats packages and equation solvers. Derive 2.0 runs under DOS and costs £130 from Chartwell-Bratt on 081 4671956.

X terminal for Microsoft Windows

XVision V4.0 allows a PC running Microsoft Windows to function as an X terminal without leaving the PC environment. A network of PCs using the TCP/IP protocol may be connected to any number of UNIX or non-UNIX hosts to enable the displaying of multiple X applications together with local Windows applications. XVision costs £399. Contact VisionWare on 0532 788858 for more information.

Spiritual Healing

A new software testing tool called Ghost from Vermont Creative Software enables developers to carry out comprehensive regression testing quickly and easily. Because it traps and records keystrokes at the DOS interrupt level, it is independent of the language used to create the executable. Ghost costs £99 and is available from The Software Construction Company on 0763 244114.



Borland swallows Ashton-Tate

In a move that has shocked the software world, US giant Borland International has announced that it is to acquire Ashton-Tate, manufacturer of the dBASE range of languages. At a joint press Borland/A-T press conference, Borland chief (and jazz flute-player) Philippe Kahn explained that, following three weeks frenzied negotiations, a deal had been reached by which A-T would become a wholly-owned subsidiary of Borland.

Borland also demonstrated two forthcoming products, tentatively scheduled for release in Q1 of 1992. Paradox for Windows will continue the expansion of the company's range of Paradox-compatible utilities, which already includes the Quattro Pro spreadsheet, the Sidekick desktop program and the Paradox engine. Object-orientation will be a major feature of Paradox for Windows with developers able to attach methods (implemented in PAL) to form objects, such as buttons. Object dBASE (formerly referred to as 'Turbo xBASE'), first mentioned in June's .EXE, will be a full MS-DOS and Windows development system. The product will run under Windows and incorporate a compiler and interpreter capable of producing executables for both the MS-DOS and Windows target environments. Language extensions to support Windows will be added but the compiler should be compatible with dBASE III and IV code. A fully integrated debugger will also be included.

Paradox for Windows and Object dBASE will continue to be developed in parallel and no attempt will be made in the near future to merge the two products. Speaking with regard to the dBASE and PAL languages, Borland's senior vice president, Richard Schwartz, drew the analogy between C and Pascal, saying that users should be able to choose whichever language was appropriate to their business needs. However, in accordance with Borland's commitment to interconnectivity, both file formats will be supported in each environment.

In the meantime, the future for Ashton-Tate and its product line is not altogether grim. Philippe Kahn gave assurances that the dBASE line will continue to be developed (see elsewhere in this news section for details of recent ports; also planned is a dBASE compiler, which is currently under beta-test) he dodged the question when directly asked if there would be a dBASE V. It seems likely that dBASE development as such will be wound down in favour of Object dBASE. Users will be presumably encouraged to migrate from dBASE to Borland's Windows product. It has been stated that development and support will also continue for A-T's Applause, MultiMate and Framework product lines. Most interesting is the placement of A-T's Interbase 'object' Server in Borland's strategy. The product is a multimedia, object-oriented database server. Interbase was acquired by Ashton-Tate just three months ago and is integral to Borland's vision of the future. It all looks a little too convenient not to have had some element of pre-planning. A-T's fortunes have never fully recovered from the release of the disastrous, buggy dBASE IV 1.0. Last year, A-T achieved about the same turnover with 1,700 employees as Borland did with 1,000, so A-T redundancies would seem inevitable.

Commenting on the news, George Fletcher, MD of Nantucket UK, said 'Our first reaction was that Borland had bought an expensive mailing list. However, it looks to us that Object dBASE might be the first real competition to the Clipper development path.' He admitted that it would be 'a while' before Clipper Windows would appear.

INTRODUCING THE ALL ROUND BOX

For more information or upgrade details call Borland Customer Services on 0734 321150, or complete and return the FREEPOST coupon to us today.

BORLAND C++® Everything you need for DOS and WINDOWS programming...

Now there's a vastly superior way to write Windows® applications. Borland C++. The only complete C and C++ programming environment for building DOS and Windows applications. Borland C++ comes with a complete set of tools including a Windows debugger, resource editor and compiler, and WINDOWS.H. So you don't even need the Microsoft® Software Development Kit (SDK).

...for only £299.95 with a FREE Windows Programming Book

Buy Borland C++ from your favourite retail outlet for just £299.95* and we'll send you a FREE copy of Charles Petzold's 'Programming for Windows**'. Worth £27.95 (RRP), this text is widely regarded as a must for Windows developers.

*Plus VAT **Sent on receipt of registration card. While stocks last.

BORLAND
SOFTWARE
KNOW HOW

Borland International (UK) Ltd.
8 Pavilions, Ruscombe Business Park, Twyford, Berkshire RG10 9NN Telephone 0734 320022

Borland C++ features

■ Windows support including MDI, DLL and DDE ■ ANSI C and AT&T® 2.0 C++ ■ Turbo Drive Compilers and Programmers Platform running in protected mode ■ Pre-compiled headers, increasing re-compilation speed by factors ■ Turbo Debugger for DOS and Windows ■ Whitewater Resource Toolkit ■ Turbo Profiler and Assembler

From the makers of Turbo C++,® Turbo Pascal,® Paradox,® Quattro Pro® and Sidekick.®

Please send me the FREE Borland C++ 2.0 Information Package.

NAME _____

COMPANY _____

ADDRESS _____

_____

POSTCODE _____ TEL. _____

.EXE 8/91

☐ I currently use a Borland language product.

Please return to: Borland C++ 2.0, Borland International (U.K.) Ltd, Freepost, RG1 571, Twyford, Berkshire RG10 8BR.

CIRCLE NO. 042

EXPERT for 68040

ACE Associated Computer Experts has added a new member to its EXPERT range of optimising compilers. This is a new 68040 code generator that is available for C, Fortran-77, Pascal and Modula-2. The combination of an optimising compiler with the lightning speed of the 68040 certainly adds fire to the RISC/CISC debate. The compiler costs 4800 guilders. ACE may be reached on 01031 20 6646416.

Profile Editor for OS/2

This new utility enables developers to access and modify information stored in a user, system or private profile. The editor displays a list of all the application names found on the current profile and displays the data associated with a selection. The Profile Editor costs £70+VAT to OS/2 User Group members and £100+VAT to non-members. Contact OS/2 UG on 0285 655888.

ZipitUP

This catchy product name applies to the reincarnation of a deceased piece of software called Minder. The revamped application is a software protection system that prevents illicit copying of DOS executables. Protection is made with the Core Utilities onto ZipitUP Key Disks. These have a standard DOS format plus an invisible 'lock' and are purchased from the company. ZipitUP Core Utilities cost £99.95 (but you can get an evaluation pack for £25 incl VAT and postage). It is available from Data Business on 0865 842224.

Anti-viral artillery

A new version of Dr Solomon's virus zapping toolkit (V5.0) has been released by S&S International. VirusGuard is a TSR that warns a user if an attempt is made to manipulate an infected file. CheckVirus is a utility which traps a file that has been maliciously modified by recording its checksum. There is also a FindVirus tool which can identify a given virus. The Anti-Virus Toolkit costs £59 and is available from S&S International on 081 6912735.

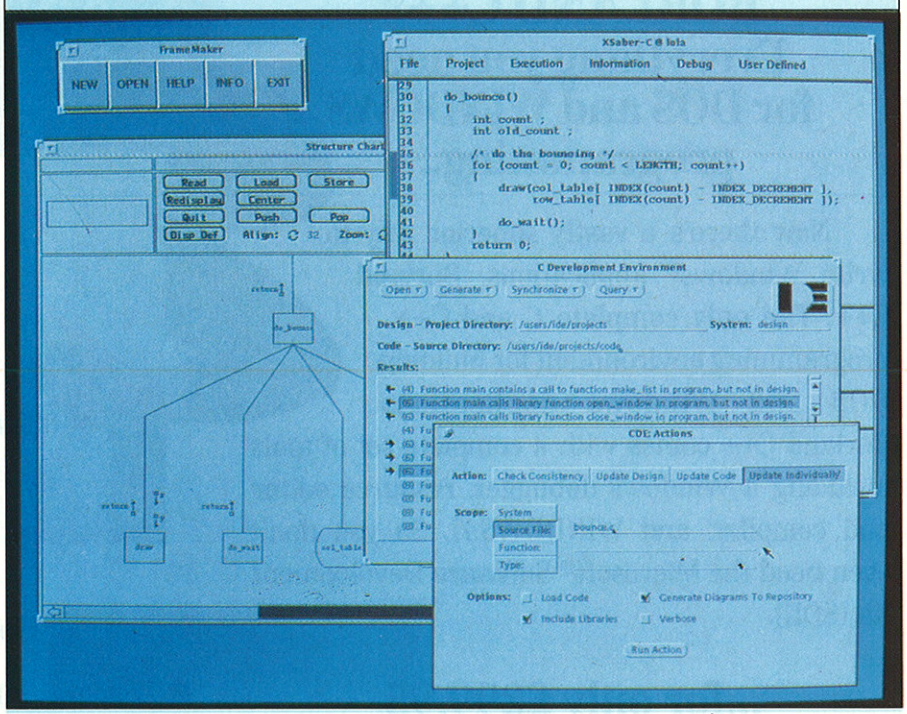
New and Improved FASTBACK

Fastback Plus V3.0 is Riva's latest release of its back-up utility program. Speed of backups and restores has now increased and there is also a 15% higher level of data compression than on the previous version. A scheduler and macro recorder enables fully automated backing up. Fastback Plus V3.0 costs £145. For more information contact RIVA Ltd on 0420 22666.

State-of-the-Art C Development

Interactive Development Environments (IDE) has announced its C Development Environment (which I shall refer to as CDE, since the company has neglected to provide an abbreviation). CDE is a tasty casserole of exotic tools for developing C software on UNIX workstations and servers. IDE's CASE tool, Software through Pictures, is integrated with the Saber-C compiler/debugger and the FrameMaker or Interleaf publishers (see pickie). Features include reverse engineering and code generation modules, facilities to synchronise code and designs, query the shared repository, and navigate among the components. CDE is designed to be beneficial even if introduced half-way through a project. The company also believes that most developers do not adhere rigidly to formal software engineering methods, so CDE does not impose any. But IDE reckons that because everything is so integrated, it will encourage people to use its CASE technology.

The C Development Environment is available from August 1 on Sun SPARCstations. IDE flatly refuses to give us any indication of how much the thing costs. Phone IDE on 0483 579000 (and see how far you get).



Clever Debugger

Convex Computer, formed in 1982 and which previously specialised in complete solutions in the supercomputing market, is now offering a source-level debugging tool that has the ability to analyse even highly optimised code. A windowing environment allows simultaneous viewing of the source code, assembler and program output. Convex is claiming that its CXdb tool is more efficient at pin-pointing complicated programming problems than other, more orthodox, debugging tools. CXdb costs £3500 and is distributed in the UK by Convex Computer Ltd on 0372 386696.

10BASE-T Networks

3Com has released a new modular hub to allow cheaper expansion of 10BASE-T networks. The MultiConnect TP Starter kit has 12 ports and can take a maximum 15 expansion modules, enabling a user to have up to 45 ports in total. Each module contains a number of LEDs which can be used to trap faults. An abundance of networking media are supported, including thick and thin Ethernet and twisted-pair cabling. The MultiConnect TP Starter Kit costs £1185 with each additional 3Comm TP Module costing £395. 3Comm can be reached on 0628 890670.

A CASE for better productivity

SE/Workplace is a Windows-based tool that provides links for the LBMS Systems Engineer CASE family into desktop applications such as word processors, spreadsheets and E-mail. It uses Dynamic Data Exchange (DDE) to transfer data from one application to another in real time without user intervention. This permits the criteria of a given query, on the Systems Engineer database, to be embedded into the selection process of any other application using the database. SE/Workplace runs under a Windows or a DOS 386/486 environment and costs £1500. It is distributed in the UK by LBMS on 071 6364213.

WELL? IT DOES EVERYTHING ELSE

Clipper is the standard for companies all over the world who are serious about selecting the correct software to develop their applications. From accounting to zoology, from finding oil to finding your way home, Clipper can help any company to do what it needs to do quicker and more efficiently. Clipper combines the power of a programming language with the convenience of a data base to offer large companies a means of saving money. And offer small ones a means of making it. However, there are some things that Clipper can't do.

We'd be straining credibility to suggest it would make the tea for you. Although out of 300,000 programmers using Clipper, there just might be someone out there . . .

Nantucket®

Clipper 5.0
The Application Development Standard

NOW SHIPPING 5.01

Nantucket UK Limited
8 Bridgegate Centre, Martinfield,
Welwyn Garden City, Herts. AL7 1JG
Telephone:(0707) 373600
Fax: (0707) 373350

I'd like a flavour of Clipper, please send me details.
Please tell me about the uses Clipper is being put to

.EXE 8/91

Name _____

Job _____

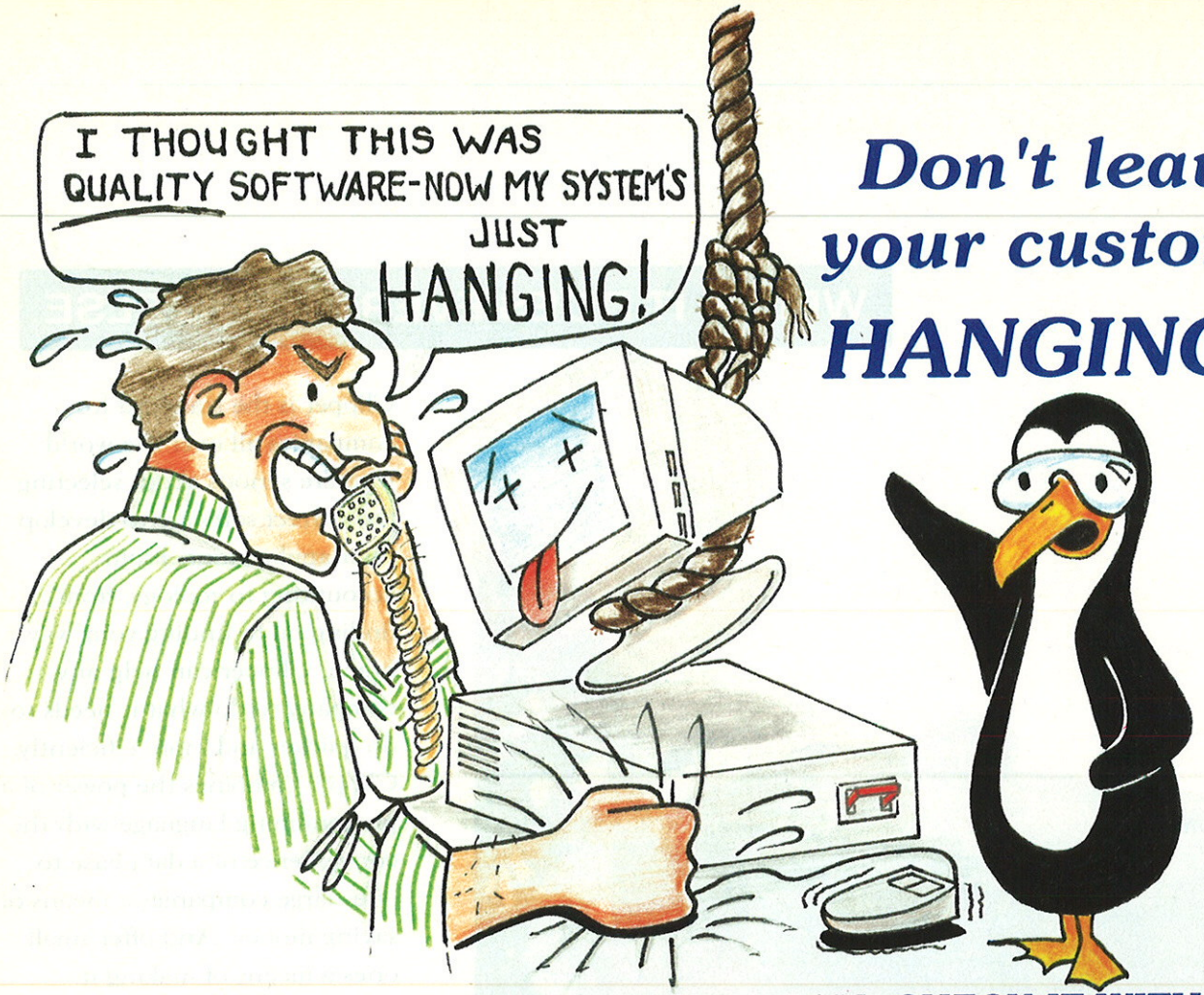
Company _____

Address _____

_____

Phone _____

CIRCLE NO. 043



Don't leave your customers HANGING. . .

BEFORE YOU SHIP ANY PROGRAM, CHECK IT WITH

COMPUTER LANGUAGE PRODUCT EXCELLENCE AWARD 1990 SOFTCOLA

BOUNDS-CHECKER™

WINNER PC MAGAZINE BEST OF 1990

JANUARY 15, 1991 PAGE 168

FINDS OUT-OF-BOUNDS MEMORY ACCESSES *AUTOMATICALLY*

Your program may have 10,000 to a million lines of code. It may occasionally hang mysteriously or it may appear to run flawlessly every time. But under DOS, how can you ever be sure that your program is not corrupting memory it does not own? The only way to be 100% sure is to BOUNDS-CHECK before you ship.

To use BOUNDS-CHECKER you build your program with debugging information (we support most compilers including Microsoft, Borland & JPI). Then you just type <BC file-name>. BOUNDS-CHECKER sets up the 386™ /i486 for protection and lets your program fly. If your program accesses memory it does not own or overwrites its own code, BOUNDS-CHECKER pops up displaying the offending SOURCE-LINE or instruction.

Programming under DOS is a gamble, so why not stack the odds in your favor--CALL TODAY.

(603) 888-2386

OTHER FINE NU-MEGA PRODUCTS. . .

The ultimate systems debugger.

Debug:
- Interrupt routines
- Device drivers
- T&SRs
- ROMS
- Applications
- Overlays

Soft-ICE™

Features:
- Break out of a hung program
- Real time Break-Points
- Back-Trace history
- Works with other debuggers

If you are debugging an application, Soft-ICE is seamlessly integrated with BOUNDS-CHECKER so you can easily go back and forth between BOUNDS-CHECKing and debugging: a combination many programming professionals can't live without.

1988 BYTE AWARD OF DISTINCTION

Run CODEVIEW for Windows on a single monitor
CV/1.....$129
• Runs in a window • No annoying flash •
• Runs on any display that supports windows •

Call by 4:00 PM EST TODAY and ask us to EXPRESS you an info packet. In most cases you will receive it by 10:30 AM tomorrow. (USA only)

All Nu-Mega products require a 386, 386SX or 486. MS-DOS and Codeview are trademarks of Microsoft Corp. 386 is a registered trademark of Intel Corp. Nu-Mega, BOUNDS-CHECKER, Soft-ICE and CV/1 are trademarks of Nu-Mega Technologies, Inc.

™ Nu-Mega TECHNOLOGIES INC

P.O. Box 7780 • Nashua, NH • 03060-7780 U.S.A.
(603) 888-2386 • Fax (603) 888-2465

BOUNDS-CHECKER $249
Soft-ICE $386

Special Offer
Buy BC & S-ICE .....Save $100

30 Day Money Back Guarantee

CIRCLE NO. 044

Lend Me Your Ears

Talking computers are nothing new, but Monologue V2.0 from First Byte seems very easy to use. Without any special hardware requirements, Monologue will speak text in a document or on the screen using the PC's own built-in speaker.

It uses a sophisticated set of phonetic translation and pronunciation rules to synthesise speech, including stress patterns (ie it doesn't drone on in a monotone). The program loads as a TSR which is activated by a hot-key combination in your editor, spreadsheet, database etc. Once you've highlighted some text you just hit RETURN and Monologue starts reciting the thing. I'll admit that it doesn't have quite the earthy resonance and crystal clarity of John Gielgud, but it ain't bad. Apparently some PCs have much better voices than others (mine was embarrassingly poor and has promised to give up smoking), so you can get various hardware add-ons to boost their performance.

There is also a toolkit available for First Byte's SmoothTalker speech driver, with an API so you can write talking DOS programs. Languages supported are C, Pascal, QuickBASIC and assembler. In addition, a character-oriented device driver is included. This may be used by any application which allows macros or languages to write to an external device (eg Lotus 1-2-3, Quattro, dBASE, Foxbase, Clipper etc).

Monologue V2.0 is priced at £89 and The Speech Toolkit retails at £395. Both products are available from Iansyst Ltd tel: 071 6075844

Microsoft Visual Basic Ships

Microsoft Visual Basic Programming System for Windows (see .EXE June '91) is now in the shops. The product is a Windows 3.0 development tool which comprises a GUI screen designer and a general purpose programming language. You simply design your front-end with the interface builder (no code necessary) and then implement event procedures in the VB language.

The VB language is a mutant offspring of Microsoft QuickBASIC, with some corrective surgery to enable it to handle the event-driven graphical environment. It uses a threaded p-code incremental compiler and comes with a source-level debugger. This allows users to create compiled Windows .EXEs that can be distributed freely without any run-time fees or royalties. Support is provided for DDE and you can extend the control set using Microsoft C, the Windows SDK and the VB Control Development Kit (available separately).

Hot on the heels of VB's arrival comes the announcement of the Visual Basic Library and SDK for Microsoft SQL Server. This adds VB to the list of languages from which the SQL Server API (DB-Library) can be called. So prospective VB programmers can access SQL Server and DB2 on the mainframe, using the Database Gateway from Micro Decisionware.

Visual Basic costs £139 but registered users of QuickBASIC and the BASIC PDS can get a copy for a mere £79. The VB Control Development Kit is priced at £39 and the SQL Server SDK costs £335. Microsoft is on 0734 500741.

386 DOS Extender Upgrade

Phar Lap has released a new DPMI compliant version 4.0 of its 386 DOS Extender. This will allow DOS-extended applications to run under Windows 3.0 enhanced mode. DPMI (DOS Protected Mode Interface - see .EXE April '91) is a new industry standard which allows protected mode DOS applications to run under Windows and future multi-tasking environments certified as DPMI compliant (eg OS/2 V2.0). It should now be possible for extended DOS programs to communicate with Windows programs via the clipboard. Phar Lap's 386 DOS-Extender V4.0 is the first extender to support all five industry standards (INT 15, VCPI, XMS, DPMI and VDS) and is also compatible with the new DOS 5.0.

Phar Lap's 386 DOS-Extender SDK V4.0 costs £325 and is supplied as a free upgrade for V3.0 users. To upgrade from V2.2d costs £145 and £25 for 386 VMM. Prices quoted are from System Science on 071 8331022.

Smooth Lines

Analog Devices/Edsun Laboratories claims that its continuous edge graphics digital to analogue converter (CEG/DAC) will dramatically improve the appearance of lines and circles on a standard analogue colour VGA monitor, by using a method of anti-aliasing. True 24-bit colour images can be displayed on an 8-bit system using dynamic palette loading. The CEG/DAC costs £49.95. For further information contact Computer and Peripheral Technology Limited on 0329 825152.

Books for us

A new bookshop has opened in London and its just for computer buffs. Books are arranged by subjects, and if perchance you're not able to tear yourself away from your PC, a mail ordering facility is also available. The PC BookShop is at 21 Sicilian Avenue, Southhampton Row, London WC1A 2QH, tel 071 8310022.

PC-MOS Source Code Licence

A Source Code Licence for TSL's PC-MOS multi-user operating system has been released. Costing $17,500, it will allow DOS distributors, VAR or system integrators to adapt PC-MOS for their own applications. Contact The Software Link on 0101 404 4485465 for more information.

Fastest 386SX from AMD

AMD has introduced a new family of SX microprocessors. The Am386SXL is a 25MHz processor that is a plug-in replacement for the 80386SX, offering less than one milliamp power consumption in stand-by mode. The Am386SXL costs $89 in quantities of a 1000. Contact Advanced Micro Devices on 0483 740440 for more information.

Conference on Software Engineering

The 3rd International Conference on Software Engineering for Real Time Systems, will be held from the 16th to the 18th of September at the Royal Agricultural College. For a copy of the conference programme and further information call the IEE Conference Services on 071 2401871 ext 325.

Access to X.400 for AppleTalk

Intergralis has released Worldtalk 400, a package that allows users of Microsoft Mail on AppleTalk networks to send and receive mail from users on X.400 systems. Users can choose their preferred E-mail package and, at the same time, get access to the X.400 services. WordTalk Microsoft Mail Edition costs £7900. For more information contact Microsoft Corporation on 0734 391123.

Entertaining Mr Sloane

dBASE IV version 1.1, Ashton-Tate's ubiquitous DBMS, is now available on Intel 80386/486-based PCs running UNIX. The product should look and feel very much like the DOS version, but as the company's UK managing director, Paul Sloane, put it: '[it] takes advantage of the multi-user, multi-tasking functionality and value-added performance features of SCO operating systems' *(sic)*. In the last six months dBASE IV versions have also been released for SunOS, VAX/VMS and Macintosh.

dBASE IV for 386 UNIX will run on five desktop systems: SCO UNIX V/386 and SCO XENIX 386, AT&T UNIX V/386, Interactive UNIX V/386 and ESIX System V. Prices start at £795. Ashton-Tate is on 0628 33123.

Letters

We welcome short letters on any subject that is relevant to software development. Please write to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked 'Not for Publication', it will be considered for inclusion on this page.

Message TO our sponsors

Sir,

.EXE does it again!

Congratulations on another issue of .EXE, with the best articles and the worst adverts of all computing magazines.

You offer a prize for the best letter - how about a booby prize for the dullest, plainest advertisement? Or, if that's too negative, what about one for the most entertaining, witty and creative ad? You could afford to offer a crate of champagne as a prize. After all, most months there would be no winner!

Andrew Gravell
Dept of Electronics and Computer Science
University of Southampton

Sounds like a great idea to me - Editor.
Hmm... - Advertising Manager.

GOTO spoof?

Sir,

I read Mark Hurst's article in the June issue of .EXE with considerable interest. I subscribe entirely to his views but was rather perplexed at the code fragment given, which seems to overturn the very opinions he had just expressed. Unless it is a subtle and clever spoof, and I just missed the joke, I have to admit that it is an ingeniously contrived way of 'justifying' a `GOTO`, not that I've ever particularly felt that it was something that needed an excuse anyway.

Perhaps we should now inform Messrs Kernighan and Ritchie that 20 years of progress has led us to a position where we can recast the 'Hello, world' program in a more readable and readily maintainable form - see Figure 1.

Graham Stokes
Warminster
Wiltshire

```
#define BLOCK_START                     {
#define BLOCK_END                       }
#define PROGRAM_KERNEL                  main()
#define DO_PRINT                        printf
#define TEXT_MESSAGE                    "Hello, world\n"
#define ESSENTIAL_PIECE_OF_SYNTAX       ;

#include <stdio.h>

PROGRAM_KERNEL
BLOCK_START
DO_PRINT(TEXT_MESSAGE)ESSENTIAL_PIECE_OF_SYNTAX
BLOCK_END
```

Figure 1 - #define world

```
#include <stdio.h>
void main(void)
{
  float p = 0.00027856;

  printf("%.1g   %.2g   %.3g\n", p, p, p);
}
```

Figure 2 - printf problems

5.1 trouble

Sir,

Is the nightmare flaw in the `printf` routine of Microsoft C, version 5.1, common knowledge? Try the program in Figure 2 and you will see what I mean. `Repeating the process with p = 0.00097856` suggests that the algorithm used is totally incorrect.

M J Healy
Harpenden
Herts

For the benefit of any readers as ignorant/forgetful as myself, the use of a '.' in a `printf()` *format specifier dictates 'the number of characters to be printed, the number of decimal places, or the number of significant digits' depending on the type character being used (Microsoft C 5.1 manual). Unfortunately, I have been unable to duplicate Mr Healy's experiment, the office copy of the C 5.1 software having gone AWOL. Borland C++ prints out* `0.00030.000280.000279`*, which seems reasonable.*

High Cast

Sir,

Can any one of the 350,000 Borland C++ users out there tell me how I can construct and index (or iterate) through an Array which contains objects that are multiply derived when the compiler will not allow me to typecast from `Object&` to `multiplyDerivedClass&`?

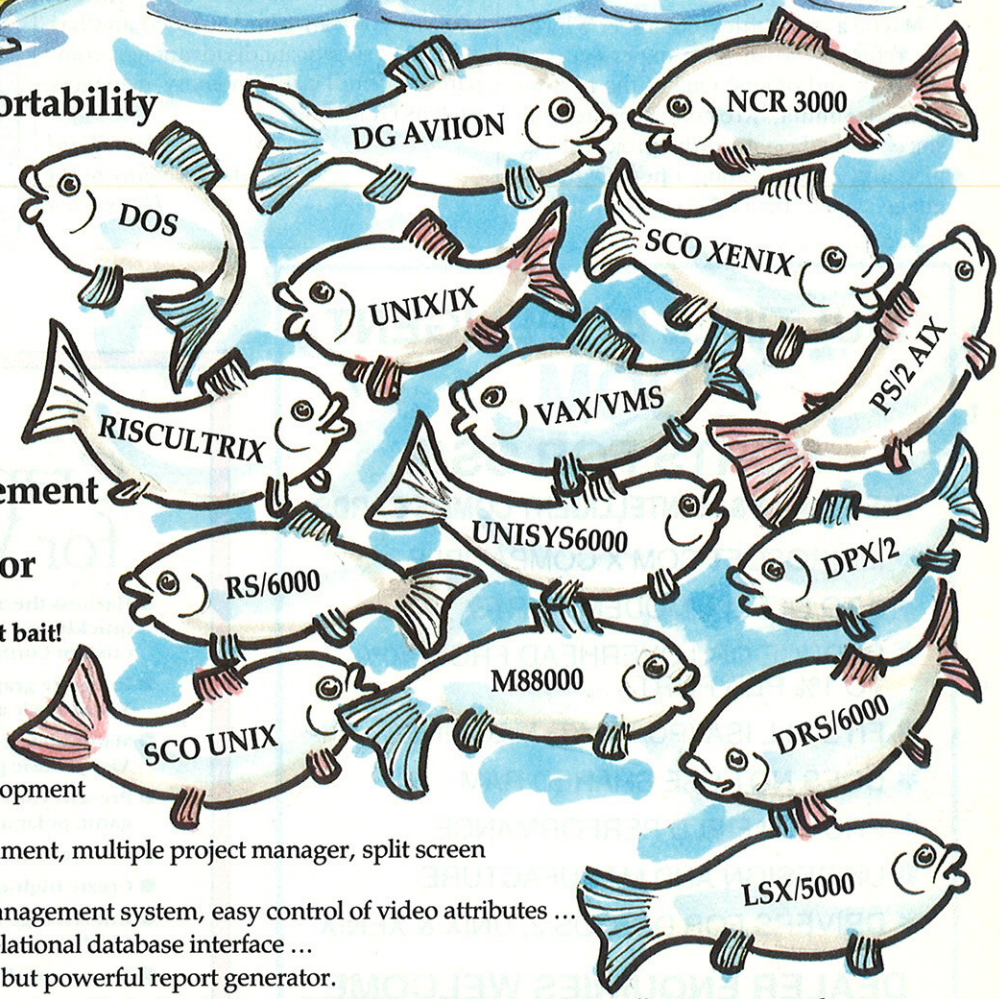
Yours in frustration++

Dave Midgley
Macclesfield
Cheshire

PS Get well soon, Verity.

Drop us a line!

- **Complete application Portability**
- **Pop up windows**
- **Pull down menus**
- **4GL RDBM's links**
- **Netware compatible**
- **Codeless screen management**
- **Powerful report generator**



To catch the right fish, you need the right bait! Ryan McFarland's latest family of RM/.. development tools together with RM's legendary portability means that we can offer you a rather special baitbox;

RM/COBOL-85, high level portable development engine, latest windowing functionality.

RM/CO, integrated development environment, multiple project manager, split screen debugging, flexible editing ...

RM/PANELS, form design and screen management system, easy control of video attributes ...

RM/Plus DB revolutionary transparent relational database interface ...

RM/Companion, due for release!! Simple but powerful report generator.

RM/COBOL-85 for Netware, provides transparent interface between COBOL application and Novell's Btrieve record management system ...

If you are fishing in the Open Systems market and you want to get your customer hooked, then drop us a line at ...

Liant Software Limited
Ryan McFarland
2 Caxton Street
St. James's Park
London SW1 0QE
Tel: 071 799 2434
Fax: 071 799 2552

CIRCLE NO. 045

Name_____
Company_____
Address_____
_____
_____
Telephone_____ Fax_____

.EXE 8/91

Simula snub

Sir,

The May issue of .EXE had an article entitled 'Why programming is hard'. A good idea to write about. And there were many good things in that article. But: The author thinks that OOP is the same as Smalltalk. That is not true. He also thinks that Smalltalk is the grand-daddy of OOP (page 59 3rd column).

I can understand that somebody not reading .EXE could make that mistake. But .EXE is the only magazine that I know which has published an article about the 'real grand-daddy' - and not so long ago at that. The November '90 issue of .EXE told the story of the 'Language that would not die' - all about Simula, the first OOP language. It was not from the US and it is not a new thing. It came from Norway, 24 years ago!

Maybe a good article could be: Why do we refuse to learn from experiences of our own and others? One of the persons behind Simula, Kresten Nygaard, has since worked on a Nordic project designing and implementing a new OOP language called Beta. I commend this to .EXE readers as a very interesting thing.

John Plate
Copenhagen
Denmark

... and QEMM snub

Sir,

It's difficult to imagine a more superficial and insubstantial review of MS-DOS 5.0 than Mark Hamilton's in the July '91 issue, but then it was only intended as an introductory article. However, it does contain a glaring factual error.

It was Quarterdeck Office Systems who were the first to exploit the High Memory Area, and not Microsoft. No way are Microsoft leaders in any form of technology, and they never have been. MS-DOS 5 doesn't compare well with DR DOS 5 (you can't load the DOS kernel into a UMB as you can with DR-DOS) and has at least 12 significant disadvantages compared to third-party memory managers such as QEMM-386.

Andrew Ward
Award Software Ltd
Harpenden

Mark Hamilton wrote his article under close supervision from this office. I felt that information on any new DIR switches in DOS 5 and graphical shells would be widely obtainable from other, less technical journals and, in the small amount of space that I had available in that issue, decided to concentrate on the API and other low-level issues.

Apologies to Quarterdeck for mis-attributing the discovery of the HMA.

I think that Mr Ward's attack on Microsoft is rather over the top (after all, Microsoft's fortunes were founded on a very innovative implementation of BASIC; there are many who believe that the company is about to score again with another). I also think that he is a bit of a tease for not listing at least some of his '12 significant disadvantages' - Ed.

Letters submitted to this page may be edited. The writer of the best letter of the month, as judged by the Editor, will be rewarded by a T-shirt or similar-valued .EXE trinket. The best letter is the one printed first.

FORTRAN 90, the new FORTRAN Standard

The new FORTRAN standard has at last been finalised, and the first compiler announced. John Reid explains the advantages of the new language definition.

It has been a long haul, but the new FORTRAN standard was finalised, down to the last editorial detail, on April 11th, and the first compiler has been announced. This seems a good moment to explain the advantages of the new standard and briefly review its history. Of course, I cannot give a complete description in an article of this length. If you want to know more, there are two books available and soon you will be able to purchase the official standard (details at the end of this article).

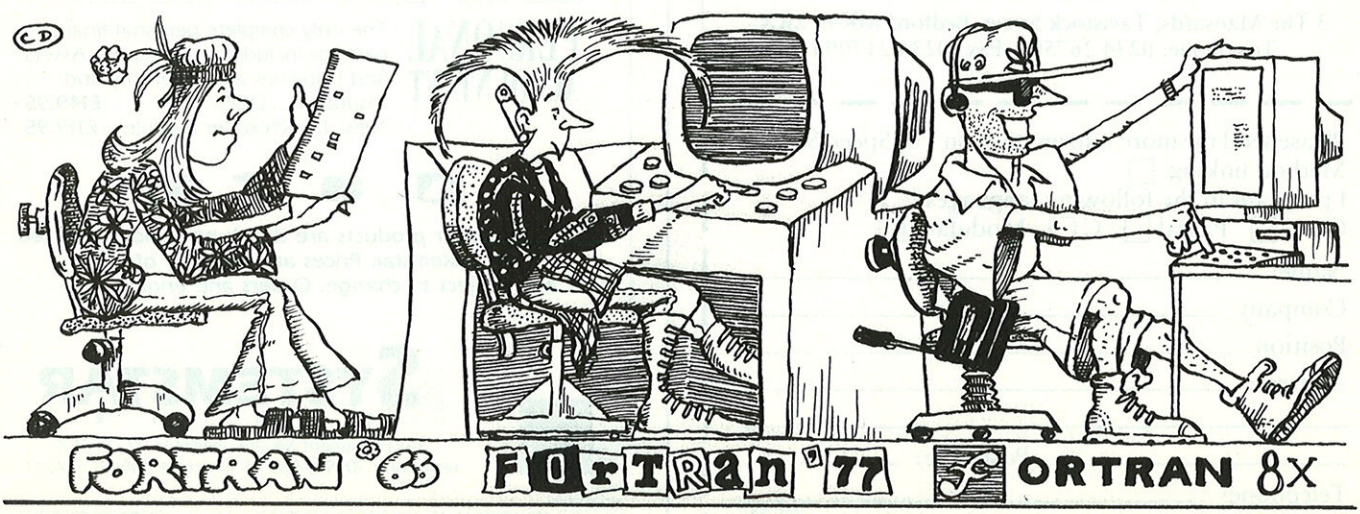
History

FORTRAN was the first computer language ever to be standardised. The original standard of 1966 was replaced by a new standard in 1978 and the languages have become informally known as FORTRAN 66 and FORTRAN 77 (it is not FORTRAN 78 because the technical content was completed in 1977). FORTRAN 77 was a modest revision of FORTRAN 66 and the ANSI committee X3J3 felt under tremendous pressure to add further features, such as dynamic storage and array syntax. It therefore went immediately into a 'tutorial' mode, learning about experiences in other languages, and aimed to produce a new standard in 1982. In fact, by 1982 most of the new features had been agreed, each as a separate extension, and the task of integrating everything into a new document began. This was when I attended my first meeting and I became a member of X3J3 at the start of 1983. The draft was then known as 'FORTRAN 8x' and I expected that the unknown x would have the value 8 at most.

The basic reason for the delay was the difference of opinion between those who wished to see a large range of new features and those with more modest goals. A ballot of X3J3 held in April 1986 was 16 to 19 against the draft of that date and led to some slimming down of the language. A second ballot held in January 1987 showed that better agreement had been reached (29-7) and a draft was issued for public comment later that year. Over 400 letters were received, representing varying shades of opinion, some welcoming the power and safety of the new features but many saying that the language was too complex and lacked certain popular extensions. The rules of X3J3 say that, at this stage, no change may be made without a 2/3 majority vote. This led to deadlock for two meetings. Finally, in September 1988, the ISO committee WG5, despairing of ever seeing a standard emerge from X3J3, defined exactly which changes it required and set a timetable for the preparation of a second draft.

```
SUBROUTINE X(N,A,B)
INTEGER N, A(N), B(N)
INTEGER WORK(N,N)
INTEGER,    ALLOCATABLE    ::
HEAP(:,:)
```

Figure 1 - Declaring dynamic arrays



```
TYPE POINT
  REAL X, Y
END TYPE POINT
TYPE(POINT) A, B(10,20)
```

Figure 2 - A derived type for points

Since 1988, progress has been steady, although there have always been about 10 'no' votes for successive versions of the whole standard - not enough to prevent acceptance by a 2/3 majority. A second draft was issued for public comment in 1989 and the 150 responses were largely favourable. A third draft in 1990 provoked only 29 letters. The new standard was finalised on April 11th. The technical content was completed last year, which is why the name 'FORTRAN 90' has been chosen (just as with FORTRAN 77, which was completed in 1978). The camera-ready copy was sent to ISO in Geneva at the end of April and the publication date is expected to be in the first week of August.

Language evolution

FORTRAN has been around for a long time and there is a huge volume of working code. To protect this investment, FORTRAN 90 is a proper superset of FORTRAN 77 - a program that conforms to FORTRAN 77 will conform to FORTRAN 90 too. This aspect has never been controversial within the committee.

Looking to the future, the committee has given everyone a warning of possible deletions in the next revision by labelling a small number of features that have replacements in FORTRAN 77 as 'obsolescent'. Many shops already advise against the use of these features and again this has not been controversial within the committee. The features involved are:-

- Arithmetic IF
- Non-integer DO index
- DO termination other than on a CONTINUE or END DO statement
- Branching to END IF from outside its block
- Shared DO termination
- Alternate return
- PAUSE
- ASSIGN and assigned GO TO
- Assigned FORMAT specifiers

Array features

The fact that all FORTRAN 77 arrays are static is a very big deficiency. FORTRAN 90 contains 'automatic' arrays, created on entry to a subprogram and destroyed on return, and 'allocatable' arrays whose number of subscripts (rank) is fixed but whose actual size and lifetime are fully under the programmer's control through explicit ALLOCATE and DEALLOCATE statements. The declarations in Figure 1 include an automatic array WORK and an allocatable array HEAP. Note that a stack is an adequate storage mechanism for the implementation of automatic arrays, but a heap will probably be needed for allocatable arrays.

These two changes represent an enormous advance from FORTRAN 77 with its static storage. There will no longer be any need for work-space to be set up by the user of a library procedure or for the argument list to be cluttered with work-space arguments. It will now be straightforward to structure global storage according to the size of the problem in hand and there will no longer be any need for complicated and unsafe storage management schemes within the code itself.

```
MODULE INTERVAL_ARITHMETIC
  TYPE INTERVAL
    REAL LOWER, UPPER
  END TYPE INTERVAL
  INTERFACE OPERATOR(+)
    MODULE PROCEDURE ADD_INTERVALS
  END INTERFACE
  :
CONTAINS
  FUNCTION ADD_INTERVALS(A,B)
    TYPE(INTERVAL) ADD_INTERVALS, A, B
    ADD_INTERVALS%LOWER = A%LOWER + B%LOWER
    ADD_INTERVALS%UPPER = A%UPPER + B%UPPER
  END FUNCTION ADD_INTERVALS
  :
END MODULE INTERVAL_ARITHMETIC
```

Figure 3 - A module for interval arithmetic

Dummy arrays may be 'assumed-shape' (take their shapes from the corresponding actual arguments). No longer will we need to specify the leading dimensions of arrays as separate arguments when calling library codes. Arrays may be of size zero, which will mean that we no longer have to write special-case code in case it happens. Arrays may be used in whole-array expressions such as

```
B + C*SIN(D)
```

The operations are performed element-by-element, that is the sine function is applied to each element of D, multiplied by the corresponding element of C, and added to the corresponding element of B. The arrays must have exactly the same shape, but scalars may be intermixed freely. Array expressions may be used as actual arguments. They may be used in whole array assignments such as

```
A = B + C*SIN(D)
```

provided the left-hand side array has exactly the same shape as the expression. Note that there is scope for a computer to exploit fully multidimensional arrays in a statement such this, whereas if it is rewritten in the form of nested DO loops, existing vectorisation techniques often vectorise only the innermost loop.

Rectangular sub-arrays, called 'sections', may be used as arrays. Examples are A(:,7) which is the 7th column of A and A(2:10:2,7) which consists of components 2, 4, 6, 8, 10 of the 7th column of A.

Functions may be array-valued. Almost all the FORTRAN 77 intrinsics (and a few new ones) may be called 'elementally' in the way SIN was called in the above example. There are many new inquiry intrinsics that return the array properties of their arguments and many new array-valued intrinsics, for example MATMUL for matrix multiplication, MAXVAL for the largest element, and SUM for summation.

Arrays of rank one may be constructed as lists of scalars and other arrays of rank one, just as in input-output statements in FORTRAN 77. An example of an array constant of size 10 is

```
(/ 21.0, 2.7, (21.0,2.0,  &
                  I=1,4) /)
```

There is a RESHAPE intrinsic function to allow arrays of other shapes to be constructed.

WHERE statements allow array assignment statements to be masked. For example

```
WHERE (A.GT.O) B=LOG(A)
```

```
INTEGER, PARAMETER :: SKIND = SELECTED_REAL_KIND(10,99)
REAL(SKIND) A
```

Figure 4 - Selecting precision

```
INTEGER, PARAMETER :: LONG = SELECTED_REAL_KIND(10,99)
INTEGER, PARAMETER :: SHORT = SELECTED_INT_KIND(5)
REAL (LONG) PI
INTEGER (SHORT) ISHORT
PI = 3.141592654_LONG
ISHORT = 12_SHORT
```

Figure 5 - Using kind parameters

causes the evaluation and assignment of logarithms only for elements that are positive. There is also a block form with an optional ELSEWHERE block.

Derived data types

FORTRAN 90 permits data to be grouped into a structure. For example, Figure 2 shows the declaration of a 'type' for the x and y coördinates of a point together with the declaration of a scalar and an array of this type. The symbol % is used to select a component; for example, A%X is the X component of A (unfortunately the C/Pascal '.' is unavailable because of its use for operators such as .GE.).

Functions may be used to define operations on these compound data types and subroutines may be used to define assignments between them. The operators may be intrinsic (for example +, *, .EQ.), in which case the existing priorities are used for the new operators, or non-intrinsic (for example, .MERGE.), in which case the priority is maximum for unary operators and minimum for binary operators.

Derived data types provide the language with a powerful form of extensibility. It means that ordinary in-line operator notation will be available for matrices, extended precision arithmetic, interval arithmetic etc.

Modules

Modules are collections of data, type definitions, and procedure definitions. For example, a module for interval arithmetic is shown in Figure 3. It contains the definition of a type whose components are the lower and upper bounds of the intervals, a procedure for adding two intervals and an interface block that tells the compiler to associate this function with the + operator.

Access to this module requires a USE statement whose simplest form is

```
USE INTERVAL_ARITHMETIC
```

To allow for possible name clashes, there is a renaming facility. Also, access may be restricted to a list of entities.

Modules provide a safe replacement for COMMON. Note that the definitions are given only once. It is more general than COMMON in that type and procedure definitions are included. It is likely that libraries will become libraries of modules instead of libraries of procedures.

Procedures

A procedure may be called recursively provided its leading statement includes the qualifier RECURSIVE. It may be internal (at one level only) to an external subprogram or to a subprogram in a module. Keyword calls, as in the I/O statements of FORTRAN 77, are available. The dummy argument names serve as keywords. Arguments may be omitted provided they are declared as OPTIONAL. The intrinsic function PRESENT may be used to inquire whether an optional argument is present. Dummy arguments may be declared to be IN, OUT, or INOUT.

Interface blocks that contain statements just like the leading statements of a procedure may be used to specify the interface to an external or dummy procedure. For example, this permits keyword calls to be made to a procedure written in assembly language. An interface block may also be used to give a generic name to a set of procedures, provided they may be distinguished by the types or ranks of their arguments. This is exactly as for the specific and generic intrinsic functions in FORTRAN 77.

Kind parameters

All the intrinsic types have been generalised to have a 'kind' parameter. This will permit processors to support short integers, very large character sets such as Japan's Kanji, more than two precisions for real and complex, and packed logicals. In the case of REAL and COMPLEX, there must be at least two kinds, corresponding to single and double precision. There is an intrinsic function that returns the kind value for a desired precision and exponent range. For example, the code of Figure 4 finds the kind value of the least precise machine representation that gives the equivalent of at least 10 significant decimals and a range of at least 1E-99 to 1E+99, and uses this to declare a real variable.

There are many inquiry and manipulation intrinsic functions that return information on the representation or manipulate parts of data (for example, extract the fractional part).

Constants may be specified with the help of an underscore and an integer constant that gives the kind value. Some examples are shown in Figure 5.

Pointers

Data objects may be declared with the attribute POINTER. Such an object does not have any storage until it is explicitly allocated for it by an ALLOCATE statement or it is 'pointer associated' with an existing target object:

```
POINTER => TARGET
```

In the case of an array, only the rank is declared initially:

```
REAL, POINTER :: A(:,:)
```

and a shape is acquired when it is associated with a target.

As a simple example of the use of pointers, suppose we have code that performs the matrix-vector product y = Ax and wish to calculate the product BCz. We might pointer associate y, A and x with r, C and z respectively, use our code to find r = Cz, then pointer associate y, A and x with s, B and r and use our code to place the result we want in s.

Components of derived types are permitted to have the pointer attribute. This permits a major application of pointers: the construction of linked lists. As a simple example, we might decide to hold a sparse vector as a chain of variables of

```
TYPE ENTRY
     REAL VALUE
     INTEGER INDEX
     TYPE(ENTRY), POINTER :: NEXT
END TYPE ENTRY
```

Figure 6 - A type for holding a sparse vector as a chain

UNIPALM TRAINING SERVICES

TCP/IP

Now an accepted standard, TCP/IP is the language of UNIX, the standard operating system now supplied by all major computer vendors. Our lab-style course gives you plenty of hands-on experience, using products such as FTP's PC/TCP, building up a fast, clear understanding of TCP/IP.

NFS

Another open systems connectivity standard, now used and accepted industry-wide. We can teach you, using products such as Sun Microsystems' PC-NFS, to understand the concepts and use and administer NFS systems effectively.

X WINDOW SYSTEM OVERVIEW

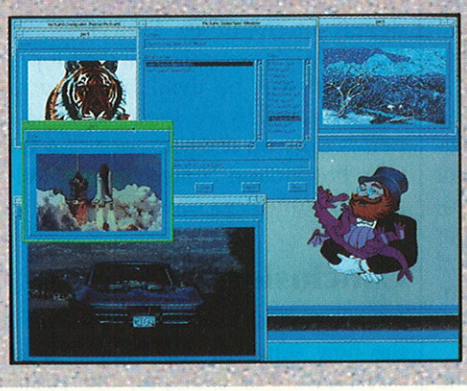
All major software vendors are thinking X now - it's playing a major part in the development of distributed applications. We used our many years of X development, support and marketing experience to develop an X Overview which gives a quick, easy understanding of the principles of X, so you can exploit its power too.

X ON PCS

To provide cost-effective additional X machines, many people use PC-based X servers to convert existing PCs to X terminals. Our PC-based X training includes plenty of hands-on experience – as user and administrator – using products such as PC-Xview from Spectragraphics, and IIM's Windows 3-compatible X server, X11/AT.

OSF/MOTIF AND OPEN LOOK

For people developing X applications, or simply wanting an in-depth understanding of the programming principles behind X, we are the only company in Europe licensed to offer the week-long OSF/Motif programming course. We can also help Sun Open Look X users to get started.



Unipalm are the only independent company in Europe licensed by the OSF to give the Programming with OSF/Motif course

To book a place on any of the courses offered by Unipalm Training Services

Tel 0954 211797
Fax 0954 211244
Intl. Tel +44 954 211797
Intl. Fax +44 954 211244

See Unipalm on Stand 200 at The Sun Show NEC 10-12 September

Unipalm Limited 145-147 St Neots Road Hardwick Cambridge CB3 7QJ England
Tel 0954 211797 Fax 0954 211244 Intl Tel +44 954 211797 Intl Fax +44 954 211244

Unipalm

CIRCLE NO. 050

the type shown in Figure 6, which allows us to access the entries one by one and create additional entries when necessary by an appropriate ALLOCATE statement. When an ordinary assignment is executed for a value of such a derived type, pointer assignment is executed for the pointer components.

To avoid performance degradation for non-pointer objects, the attribute TARGET must be declared for a non-pointer object that is to be used as a target.

There is an intrinsic function that allows enquiries to be made about whether a pointer is pointer associated and whether it is pointer associated with a given target. There is a nullify statement to disassociate a pointer.

Source form

The FORTRAN 77 limit of 6 characters in a name is raised to 31, and in-line comments following the character '!' are permitted. These two simple changes will make an enormous difference to the readability of code.

There is also a free source form that attaches no particular significance to columns 1 to 6 or 72 onwards, does not allow blanks within tokens (with a few exceptions such as END IF), and uses a terminating '&' to indicate continuation to the next line. Lines may have length up to 132 characters and statements up to 40 lines.

Miscellaneous bits

The only major new input-output features are NAMELIST and non-advancing I/O. Non-advancing I/O obviates the FORTRAN 77 insistence that records must be read as a whole and that their length be known beforehand. It is specified with ADVANCE='NO' on the read or write statement and inhibits the automatic advance to the next record on completion of the statement. On a read, if the record contains insufficient values to satisfy the input list, an end-of-record condition results and a SIZE= specifier may be used to return the number of characters read.

There are two new control structures. The CASE construct is exemplified in Figure 7.

```
INTEGER N
SELECT CASE(N)
  CASE(:0)      ! N negative or 0
  :
  CASE (1)      ! N = 1
  :
  CASE(5:7)     ! N = 5, 6 or 7
  :
  CASE DEFAULT ! Any other value
  :
END SELECT
```

Figure 7 - The case construct

```
INTEGER I
OUTER: DO                ! Unlimited DO, named OUTER
  :
  DO I=1,N               ! I=1,2,...,N
    :
    IF(...)EXIT OUTER ! Possibly exit outer loop
    IF(...) CYCLE     ! Skip to end of loop
    :
  END DO
END DO OUTER
```

Figure 8 - The DO construct

There is also a form of the DO loop that does not use labels, exemplified in Figure 8.

The MIL-STD bit intrinsic functions have been added (and made elemental).

Binary, octal, and hexadecimal values are permitted in DATA statements and there are edit descriptors for them.

Compilers are required to be able to detect the use of constructs that do not conform to the syntax of the language or are obsolescent.

Implementations

The last few years have seen a steady increase in the number of people attending X3J3 meetings, mainly from the vendors. Many of them have been actively working on implementations, but are unwilling to commit themselves to dates for release of compilers. The Numerical Algorithms Group Ltd (NAG) in Oxford, has just launched a full ISO-conforming FORTRAN 90 compiler. It has already been used to check all the examples in my book with Mike Metcalf, and is being used by NAG to begin development of its FORTRAN 90 library. NAG's compiler is a development tool and will be available on a wide range of hardware types, since it depends only on the presence of a C system. Production compilers are likely to appear from 1992 onwards.

Conclusions

I have aimed to give you a flavour of how the new standard will benefit users of FORTRAN. For example, the additions will make libraries much more friendly to use. Optional arguments, dynamic storage, and assumed-shape arrays will mean that the user need only specify what is truly special to the particular problem; and arguments may be grouped logically together into a derived type. In one case, I found that 73 arguments could be reduced to 6. If you want to have the power and safety of the new features, without the need to rewrite your code in another language, put pressure on a vendor whenever you get a chance.

EXE

John Reid is a Senior Research Scientist at the Atlas Centre of the Rutherford Appleton Laboratory. His main research interest is in the treatment of sparse matrices and he has been writing general-purpose software in FORTRAN since the 1960s. He has been a member of the X3J3 (FORTRAN) committee since 1983 and was its secretary from 1987 to 1990.

Copies of the standard (ISO/IEC 1539:1991) can be ordered from BSI Sales Dept, BSI, Linford Wood, Milton Keynes, MK14 6LE or directly from ISO, Publications Dept, Case Postale 56, 1211 Geneva 20, Switzerland. The ISO publication date will probably be August and the BSI date is likely to be a few months later.

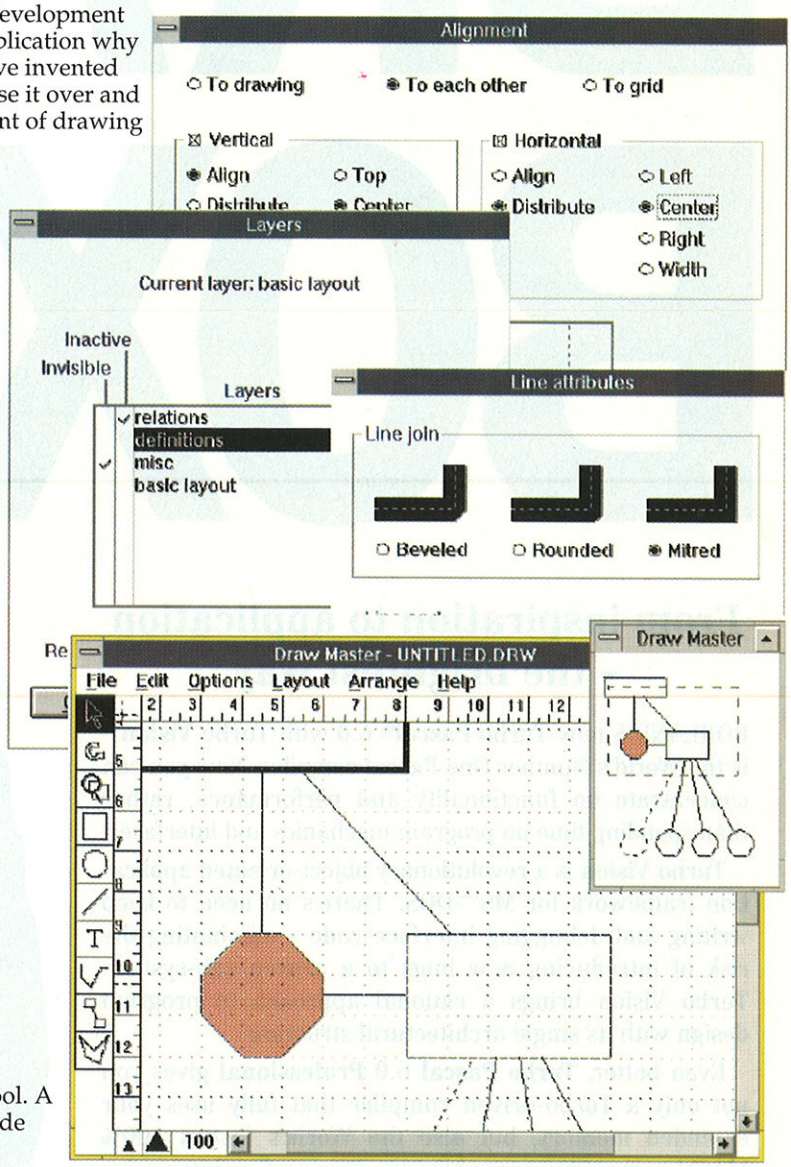
An informal description of the whole language is contained in FORTRAN 90 Explained *by Mike Metcalf and John Reid, Oxford University Press, ISBN 0-19-853772-7.*

A book that covers the principal new features and contains more examples is Programmer's Guide to FORTRAN 90 *by W S Brainerd, C H Goldberg and J C Adams, McGraw-Hill, ISBN 0-07-000248-7.*

For information about NAG's FORTRAN 90 system, contact NAG Ltd, Wilkinson House, Jordan Hill Road, Oxford OX2 8DR.

Will DRAW_MASTER save you a lot of time and money too?

DRAW_MASTER is a construction kit for the easy development of drawing tools. If you need to build a drawing application why should you reinvent the drawing tool wheel. We have invented the wheel for you and you can easily adjust it and use it over and over again. DRAW_MASTER makes the development of drawing tools easy and fast.



Product of the _MASTER Series

DRAW_MASTER has been built upon a set of classes which form an extension to the GUI_MASTER class tree. GUI_MASTER (Program Constructor for C++) is an object-oriented development environment for OS/2 PM and Microsoft Windows. GUI_MASTER consists of an Interface Builder for easy definition of objects and their properties, a class tree with over 85 classes and a class browser.

High Functionality

DRAW_MASTER consists of a class tree of more than 120 classes. Together these classes form a drawing application which offers a lot of standard functionalities such as undo/redo, copy/paste, zoom in/zoom out, alignment, patterns, drawing size. In addition it has features like layer control, overview window and resizing. When you build an application you can use DRAW_MASTER as the basis and define special features for your new application. With DRAW_MASTER you can build a new drawing tool in a few weeks.

Source code included

To give you the best benefit of DRAW_MASTER we included the source code. A cookbook and a class reference manual are also included.

Prerequisites

To use DRAW_MASTER you must have a C++ 2.x compiler and the SDK for OS/2.

Free drawing tool included

If you order now you will receive our OS/2-Draw tool. A powerful drawing tool in the OS/2 environment made with DRAW_MASTER.

Prices

DRAW_MASTER source code license $15,000.=

FEATURES

- Patterns
- Object selection
- Printing
- Multiple undo/ redo
- Changeable palette
- Rotation
- Zoom in/ Zoom out
- Polygon
- Connections
- Scrolling
- Auto scrolling
- Rulers
- Grid
- Reshaping
- Fonts
- Background/ foreground
- Grouping/ ungrouping
- Automatic Page numbering
- Page breaks
- Polylines
- Layering
- Colouring
- Keyboard shortcuts
- Resizing
- Copy/paste
- Duplication
- Dashed Lines
- Line width
- Line joins
- Line ends
- Grid snapping
- Overview
- Alignment
- 135 changeable classes
- Reuse of code
- Adding new classes
- Class reference
- Cookbook

Request for Information

Vleermuis Software Research bv
P.O. Box 2584
3500 GN Utrecht **Fax: Intl. +31 30 31 04 26**
The Netherlands **Phone: Intl. +31 30 32 49 44**

Name _____

Company name _____

Department _____

Street _____

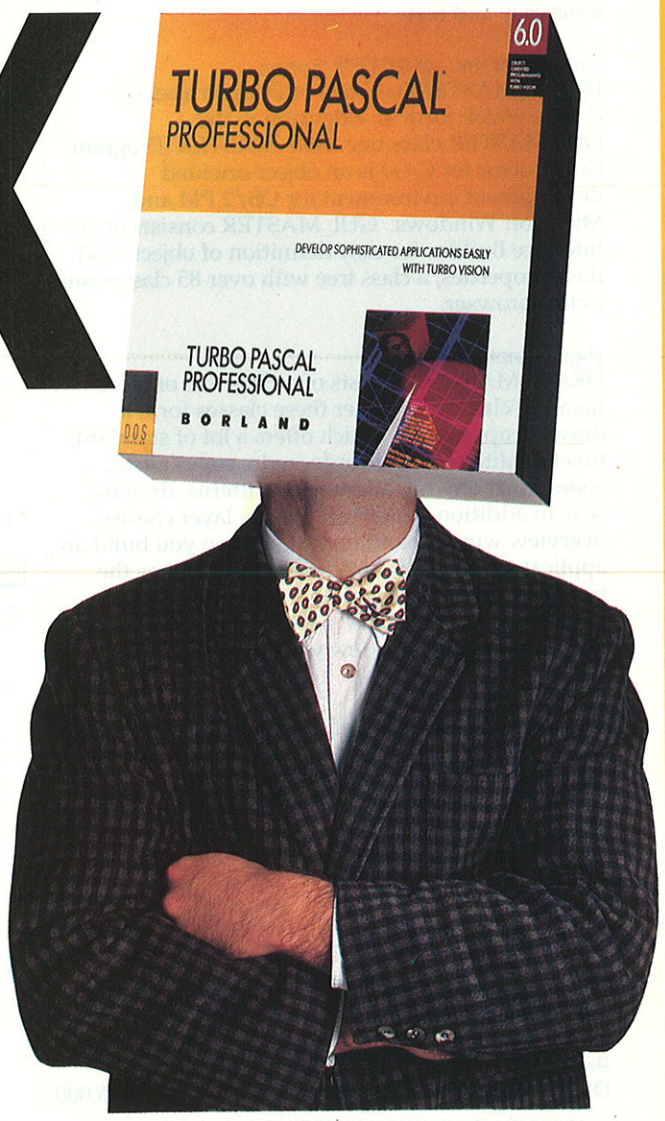
City _____ State _____ Zip _____

Country _____

.EXE 8/91

CIRCLE NO. 051

BRAIN BOX



From inspiration to application – the brightest way

BORLAND'S new **Turbo Pascal® 6.0** with **Turbo Vision™** is the World's Number One Pascal compiler. Now you can concentrate on functionality and performance, rather than spending time on program mechanics and interfaces.

Turbo Vision is a revolutionary object-oriented application framework for MS™-DOS. There's no need to keep writing and debugging interface code – eliminating the risk of introducing new bugs to a proven sub-system. Turbo Vision brings a rational approach to program design with its single architectural structure.

Even better, **Turbo Pascal 6.0 Professional** gives you not only a Turbo-driven compiler that fully uses your extended memory, but also the World's fastest 100% MASM-compatible assembler, the World's first interactive profiler, and **Turbo Debugger®2.0**, which supports OOP.

If all that doesn't make you think, think about this too! Turbo Pascal 6.0 Professional retails for a remarkably competitive £199.95 plus VAT, and Turbo Pascal 6.0 for £99.95 plus VAT. And that's not all – if you're an existing Turbo Pascal user, take advantage **now** of Borland's very special upgrade offers.

For more information, contact your Borland dealer, ring Borland Customer Services on 0734 320022, or send the Freepost coupon today.

You'll never make a smarter move!

BORLAND
SOFTWARE
KNOW HOW

Borland International (UK) Ltd
8 Pavilions, Ruscombe Business Park
Twyford, Berkshire RG10 9NN Telephone 0734 320022

CIRCLE NO. 052

From the makers of Paradox®, Quattro® Pro, Sidekick®, Turbo Pascal® and Turbo C++®

NAME _____

JOB TITLE _____

COMPANY _____

ADDRESS _____

_____

POSTCODE _____ TEL _____

.EXE 8/91

I currently use Turbo Pascal ☐

Please return this coupon to: Turbo Pascal 6.0 Information, Borland International (UK) Ltd, FREEPOST RG1571, Twyford, Berkshire RG10 8BR

Three COBOL Jewels?

Vincent Kilcoyn feels that COBOL has been neglected. Here he looks at some modern COBOL extensions and three compilers which provide them.

Over the last decade many advances have been made in programming languages. Pascal and C++ are rarely out of the headlines. COBOL seems to have been left behind. In an attempt to rectify the language's somewhat stale image, manufacturers have come up with some highly viable versions of COBOL which incorporate, to some extent, what a modern programming language should provide.

Three such versions are Microsoft Cobol from that eponymous party, RM-Cobol from Ryan McFarland and AcuCobol from the comparatively unknown AcuCobol Inc. All three can be rated as excellent compilers catering for the MS-DOS environment, and this is the platform used as the basis of comparison in this article. (The evaluations were carried out on a 386SX PC running at 20MHz with a 17ms Hard Disk.) It is worth noting, however, that all the compilers support other machines/OSs. Microsoft's compiler comes complete with full OS/2 support, and both AcuCobol and particularly Ryan McFarland are portable between a wide range of larger platforms, offering the possibility, for example, of doing development work on PCs without consuming the departmental mini's precious CPU time.

In common

All three compilers fully support the ANSI-85 standard (more on this later), and also provide a number of language extensions. These extensions are mainly in the area of file handling and windowing.

Happily, the file handling extension, which provides the programmer with the facility to read backwards through an indexed file, has been implemented in the same way by all three compilers. The syntax for this extension is `READ <filename> PREVIOUS RECORD`.

Another facility provided is the ability to use the `START` verb with a `LESS THAN` or `LESS THAN OR EQUALTO` qualifier. This statement provides the programmer with an ideal tool to access, for example, the last transaction on an account. The addition of these two extensions fills in

```
IF linecount >= 55 THEN
      MOVE 0 TO linecount
      ADD 1 TO ws-page-no
      PERFORM Newpage
END-IF
```

Figure 1 - IF...END-IF statement

the last gap in COBOL's already comprehensive file management facilities - facilities which, in the opinion of the writer, are still unmatched by any other programming language in use today.

But the windowing extensions must be the most dramatic enhancements. The modern COBOL programmer is now able to impress both his existing users and potential customers with flashy forms and clever menus. The days of struggling with cumbersome DISPLAY statements to try and put something half-decent on the screen are gone.

Unfortunately, with these extensions, the manufacturers have reverted to type: all three compilers have taken a different tack in implementing windowing. RM and ACU have extended the DISPLAY verb to include a window option, but the syntax is not the same. Microsoft offers several different methods, using add-on tools such as a dialog manager.

ANSI-85

Let us look for a moment at the features introduced by the ANSI-85 standard. The main impact of ANSI-85 is to improve program structure. Many statements can now be delimited with a suitable 'END' terminator, for example IF...END-IF. Because of these, the programmer now has the facility to produce much more readable code (see Figure 1).

This also allows the programmer to cater for nested IFs (Figure 2) - a daunting task indeed when using the old syntax. The language also now boasts an EVALUATE statement, equivalent to Pascal's case construct, which can be used to replace some repeated IF structures. There is also an in-line PERFORM - see Figure 3.

Programmer's WorkBench took approximately 20 seconds to check a small program, and the same again to reload itself

The CALL statement makes a large number of callable subprograms a viable proposition. Each subprogram has a very clean interface defined in a LINKAGE SECTION, somewhat reminiscent of Modula-2's module headers. An improvement in performance is provided by the introduction of sophisticated memory management algorithms within the environment.

Taking all the above into consideration, one must come to the conclusion that development is now far superior to that provided by the older implementations of the COBOL language.

```
IF ws-month = 2 THEN
    IF ws-days = 28 THEN
          MOVE "Y" TO ws-days-ok
    ELSE
        EVALUATE ws-days
            WHEN 29
                IF ws-leapyear = "Y" THEN
                    MOVE "Y" TO ws-days-ok
                ELSE
                    MOVE "N" TO ws-days-ok
                END-IF
            WHEN OTHER  MOVE "N" TO ws-days-ok
        END-EVALUATE
    END-IF
END-IF
```

Figure 2 - Nested IFs

Microsoft COBOL

The Microsoft compiler arrived in a huge box, raising my hackles about the number of manuals that would have to be assimilated before installation. The size of the manuals was also, unfortunately, reflected in the amount of hard disk space required to accommodate the compiler and its flock of add-ons, which together consumed approximately 6 MB of my hard disk. This reflects the comprehensive nature of the package, which contains a debugger, Programmer's WorkBench and linker, to name but a few.

After spending some time digesting the manuals, one could be quite confident in approaching a new program, especially if armed with some experience in the IBM COBOL environment. This COBOL is ideally suited to IBM mainframe dialects and even provides an old favourite of those who have had MVS experience: Dialog Manager.

Both the syntax of the language and general approach are very much in keeping with IBM COBOL - unsurprising, since the software is based on a rebadged Micro Focus compiler - this latter being an IBM favourite. The compiler generates .EXE files which devour disturbing quantities of memory (especially when working within the confines of an MS-DOS based PC). However, the debugging facilities provided by the Animator tool must be admired - they are second to none.

One of the tools which was provided with the compiler was the Programmer's WorkBench. While this provided such facilities as code checking and integration of an editor with the compiler, it must, in my opinion, be regarded with some indifference due to its slow response times. It took approximately 20 seconds to check a small program, and the same again to reload itself.

Not really the type of thing one expects of a PC-based package. The generated code itself did prove to run quite quickly, mainly due to the compiler generating native object code (the other systems use a semi-interpretive p-code system).

RM-Cobol

This product proved to be substantially different from Microsoft's: there were far fewer manuals, and the software occupied much less disk space. The installation proved to be quite simple and the learning curve shallow. When I had the system up, I immediately noticed a

When your system has a weak point . . .

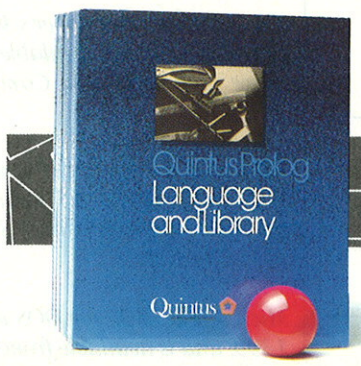
. . . Quintus Prolog is the strength you're looking for.

Today's practical systems programmers require robust, powerful and productive software tools and more than just one.
No one language offers a complete solution in itself. For example, C is a general purpose programming language offering great portability, whilst Prolog gives enormous programmer productivity for searching databases, manipulating strings, reasoning and matching patterns. Each language is powerful when used individually. When used together their potential is unlimited.

Using each language for what it does best is a good solution to the development efficiency problem. With **Quintus Prolog Release Three** complete integration can be easily achieved using C, C + + and other languages.

You can add power and sophistication to applications without adding reams of code. We can prove it.
Call us, on (0923) 211657.

So avoid weak spots, use the right tools for the job – Quintus and LPA Prolog Products available from A I International Ltd.



ARTIFICIAL INTELLIGENCE INTERNATIONAL LIMITED
Gresham House, 53 Clarendon Road, Watford, Herts. WD1 1LA
Telephone: 0923 211657 Fax: 0923 247836

CIRCLE NO. 053

marked difference in the amount of memory required by compiled programs. The compiler does not go all the way to executable code, which allows RM to reduce memory usage considerably.

Some of the features incorporated in the package reflect Ryan McFarland's many years in the Open Systems market; in particular the high degree of compression achieved on their files and the performance of the programs, especially in compute-bound applications. Personally, my biggest stumbling block with the software was the marked absence of any debug facility.

RM offers a range of add-on tools to support its compiler, namely RM-Panels and RM-Co*. RM-Panels provides comprehensive forms-development facilities in a well-presented environment.

The only down side of it is that the use of RM-Panels reduces the portability of software between compilers (although RM itself does offer a wide range). The RM-Co* product is aimed at being a programmer's workbench type thingy; it provides library handling facilities and other 'goodies'. Once again, however, I found it rather cumbersome to use and did not feel that its merits outweighed its drawbacks.

AcuCobol

AcuCobol is a comparative newcomer to the COBOL development scene, although this did not reflect negatively on the quality of the product. The packaging is very much to the point: there is only one manual and one disk.

Needless to say, installation was a very simple procedure, requiring only a brief glimpse at the manual. Once I had become familiar with the syntax for running the compiler, code development proved to be simple - and very similar to RM-Cobol. In fact, the compiler offers good compatibility with RM - theRE is even a compiler switch for RM compatibility (and another one for DEC VAX COBOL).

On compilation, AcuCobol won over the other packages both in terms of compilation times and the amount of storage required by the compiled code. The performance of the code was satisfactory (although generally the slowest of the three), and the compiler definitely 'preferred' I/O bound applications (which is, after all, what most COBOL programs are about).

The high memory usage of MS-COBOL makes the design and coding of medium to large systems unwieldy

AcuCobol lacks such frills as form generators and workbenches, though I dare say that these will be along in the future. Despite this no-frills approach, the product does provide an excellent debugger, and a file handling utility called 'VUTIL' which provides the programmer with facilities such as download, check integrity, rebuild index etc. It also contains an unusual convert option, which allows the user to convert RM-Cobol files and VAX RMS files to VISION files as used by the AcuCobol environment.

Personal Favourite

Despite the difference in performance between RM-Cobol and AcuCobol when put head-to-head on compute-bound and I/O bound problems, my preferred development environment is AcuCobol. This preference is based on a number of properties:

- Very Fast Compiler,
- Excellent Debugger,
- Compatibility,
- File Handling Utility, and
- Processing Speed.

Conclusion

In light of the above discussion it is necessary to view each compiler on its own terms; the 'Horses For Courses' *cliché* springs to mind. The languages in their skeletal forms are by and large compatible with one another, but when you start using the various extensions, this compatibility obviously tends to weaken.

One must analyse the type of application which is to be built and, in the light of such factors as environment; processing base; I/O or compute-bound; programming background; possible future trends and cross machine portability, decide on the relative merits of each system.

When the destination platform is an IBM-type mainframe, and is to be developed and maintained by people within that environment, then the choice in my opinion would be Microsoft COBOL. This is because of the similarity between MS-COBOL and IBM COBOL, and the fact that high memory usage by the programs makes the design and coding of medium to large systems unwieldy given the restricted resources of MS-DOS.

If the application is destined for either a PC or an 'Open System' environment then the choice should be either RM-Cobol or AcuCobol. The choice between these two should, in my opinion, be dictated by the nature of the processing; if the system is compute-bound then opt for RM-Cobol, otherwise choose AcuCobol.

EXE

Vincent Kilcoyne works for Solomons Stockbrokers in Dublin, Eire.

The AcuCobol Developer's kit for MS-DOS costs £950, and is available from the Software Construction Compamy (0763 244114).

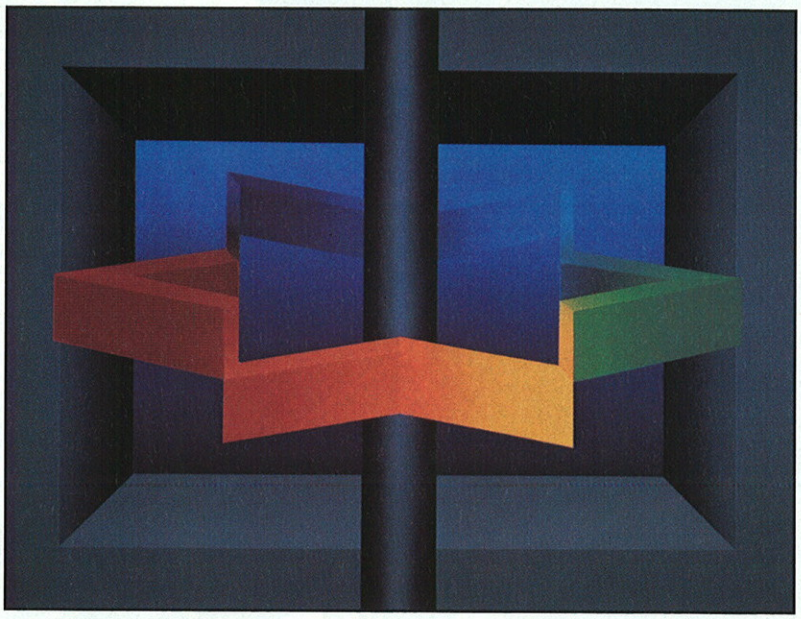
RM-Cobol for MS-DOS (bundled with RM-CO and RM-PANELS for a promotional period) costs £995 from System Science (071 833 1022).*

Microsoft COBOL for DOS and OS/2 costs £399, and is available from most dealers.

```
PERFORM VARYING sub FROM 1 BY 1
      UNTIL sub > 99
          MOVE 0 TO ws-value(sub)
          MOVE 0 TO ws-qty(sub)
END-PERFORM.
```

Figure 3 - An inline PERFORM statement

Who put C++ to Work?



glockenspiel
class constructors

Glockenspiel C++

Glockenspiel C++ leads the field in object-oriented programming. It's the most effective and efficient implementation around. It conforms exactly to the AT&T C++ specification – and they don't come any closer than that! Portable too: it's on more platforms than any other C++ implementation.

Glockenspiel CommonView®

Glockenspiel CommonView is the leading C++ class library for developing Windows, PM and OSF/Motif applications. Logical, well structured and easy to use, CommonView is 3 to 5 times more productive than the API.

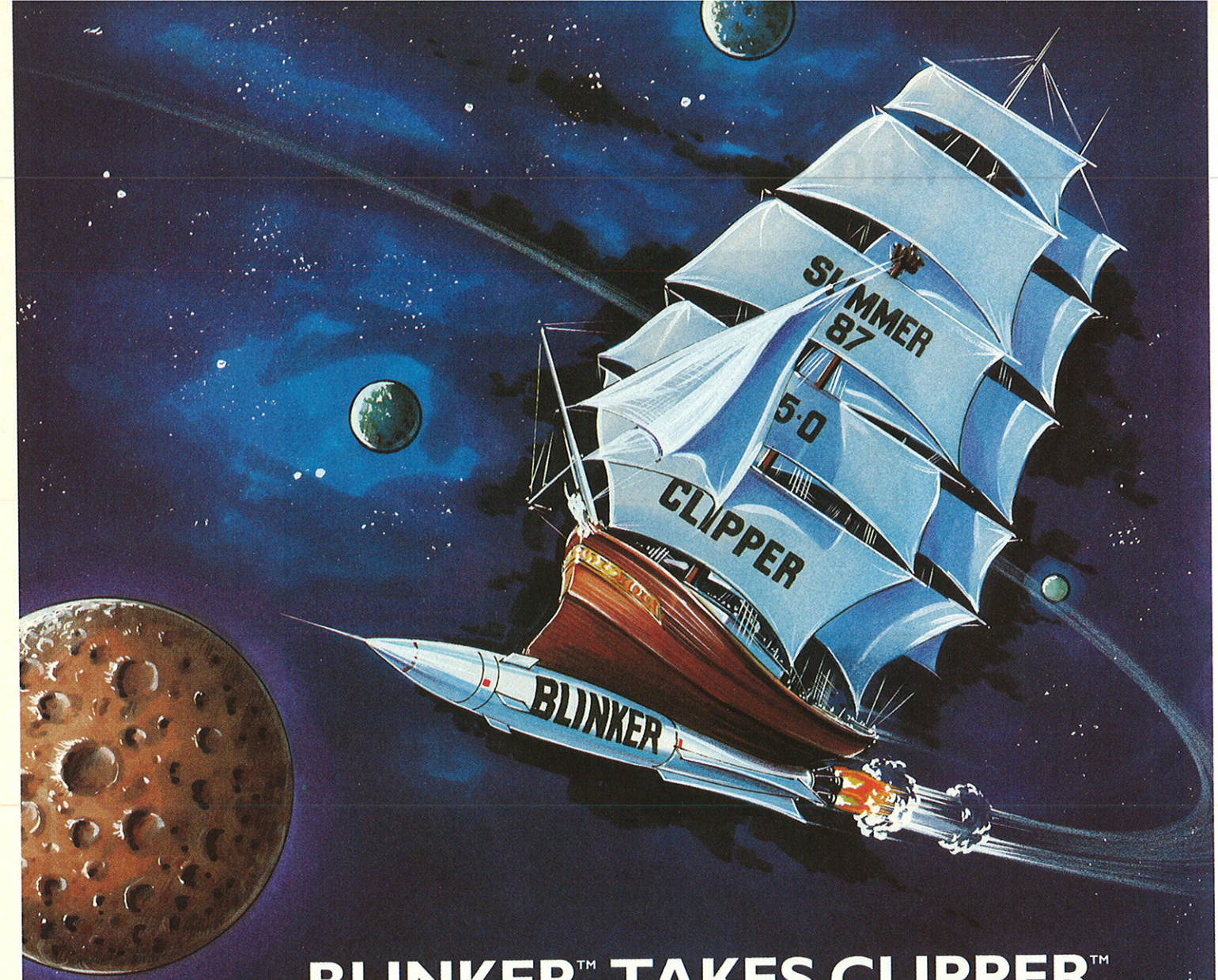
C++ Training & Consultancy

Glockenspiel C++ training courses reflect our experience as C++ developers. From design to development, *learn from the best!*
Our C++ consultancy services match your needs as you commit more resources to C++ and object-oriented development.

Further Information

GLOCKENSPIEL, 39 Lower Dominick Street, Dublin 1, Ireland. +353 (1) 733166. Fax +353 (1) 733034

■ **US: IMAGESOFT**, (516) 767-2233. Fax (516) 767-9067. **OASYS**, (617) 862-2002. Fax (617) 863-2633 ■ **UK: QA TRAINING LTD.**, (0285) 655888. Fax (0285) 650537.
■ **Italy: INFERENTIA**, (02) 26680568. Fax (02) 2364258. ■ **Sweden: LINSOFT**, (01) 3124780. Fax (01) 3152429. ■ **Germany, Switzerland, Austria: PSI**, (06021) 492-0. Fax (06021) 492-112
■ **Benelux: RIJNHAAVE**, +31 (71) 218121. Fax +31 (71) 216118



BLINKER™ TAKES CLIPPER™ OUT OF THIS WORLD

Blinker eliminates the need for complex overlay structures for both Clipper Summer '87 and 5.0

* Significantly reduces run-time memory requirements
* Accelerates your development cycle 3-10 times faster than any other current linker and even faster when linking incrementaly
* Dynamically overlays C and Assembler code sich as Extend.lib, dGE, FUNCky, other Clipper Add-ons and your own 'C' code
* Optimises your application with advanced profiling functions
* Includes memory defragmentation, 'burning-in' of Clipper environmental variables/serial numbers and creation of demonstration versions

IT'S ALL PLAIN SAILING WITH BLINKER

What the experts say

Here are just some comments from the 5000 happy users to date

"It's blinking fast!"

"Congratulations – we gained over 100k"

"It's so good, I went through all my programs and re-compiled/linked just for the pleasure of seeing all the free memory"

"It's fast! Handles overlays like a dream"

"Amazing – No more messing about with overlay structures!"

"Brilliant!... But my coffee breaks have been shortened!"

"Actually exceeded our most optimistic performance hopes! (Just for once)"

All known registrations for free update now shipped.

Blink Inc Limited, TCL Centre, Cwmbran NP44 3XR

European sales and support:

QBS Software Limited

10 Barley Mow Passage, Chiswick, London W4 4PH

Telephone 081-994 4842. Fax 081-994 3441. BBS 081-747 1979

Writing C++ Member Functions in Assembly Language

Zortech's C++ manual discourages punters from writing C++ member functions in assembly, but Laine Stump is not one to heed such warnings when there are bytes to be saved and clock cycles to be gained...

Most of you have had some experience with (or at least exposure to) linking assembly functions to your programs using the naming and parameter passing conventions of C, or 'C linkage'. While C++ programs can call these functions too, sometimes it is necessary to write assembly language functions which have C++ linkage. For example, if you want to write a member function in assembly, it must have C++ linkage.

In this article, I'll show you how to write assembly functions with C++ linkage and, in particular, C++ member functions. I have used Zortech C++ V3.0 and MASM V5.1 for my experiments and examples. Turbo Assembler should work fine in place of MASM, as long as you use the /jMASM51 directive. At the end of the article, I'll give a few pointers on interfacing with Turbo C++.

```
;; OUTBYTES.ASM
;; a function which can have either C or Pascal Linkage
;; depending on what is given in the .MODEL
;;
;; to call from C, assemble with
;;      masm /mx exam;
;; #include "outbytes.h" in your program, and call as:
;;      OutBytes(port, &vals, ct);


;; output ct bytes of data at vals to port
  .MODEL  SMALL, C ; use PASCAL to call from Pascal
  .CODE
  .186             ; assume we will run on at least 80186
;;
;; void OutBytes(int port, char *vals, int ct)
;;
OutBytes PROC  USES SI DS, \
         port : WORD, vals : PTR BYTE, ct : WORD
  MOV    DX, port
IF (@datasize)
  LDS    SI, vals  ; if far data model (C or L)
ELSE
  MOV    SI, vals  ; if near data model (S or M)
ENDIF
  MOV    CX, ct
  CLD
IF (@Cpu AND 2)
  REP    OUTSB     ; only works on 186 & > CPUs
ELSE
OB_00:
  LODSB      ;simulate REP OUTSB on 8086 and 8088
  OUT    DX,AL
  LOOP   OB_00
ENDIF
  RET
OutBytes ENDP
END


// OUTBYTES.H - prototype for OUTBYTES.ASM


extern "C" void OutBytes(int port, char *vals, int ct);
```

Figure 1 - An assembly function with C Linkage

Much of the following information was obtained not from a manual, but from hours of studying disassembled code with a debugger. Although everything here is correct for the version of Zortech C++ I am using (V3.0), Zortech does not guarantee that it will work in any previous or future versions. It reserves the right to change the C++ linkage convention if necessary 'to accommodate future enhancements in the C++ language'.

Why Bother?

As it is possible to call functions with C linkage from a C++ program, you may question the utility (sanity?) of writing assembly functions with C++ linkage. The Zortech manual even suggests that you avoid writing assembly with C++ linkage, and use C linkage; if you need to have assembly language in a member function, write the member in C++ and have it call an assembly function with C linkage.

This will work, but it will defeat much of the point of writing in assembly language. The program will be larger because there will be two functions instead of one. It will be slower for the same reason - in order to get to the code you want to execute, you must push arguments onto the stack and call the member function, which will push the same arguments onto the stack in another location, and call the assembly function. Not a model of efficiency.

C++ Complications

Writing an assembly language replacement for a C++ member function is a bit more complicated than writing a function with C linkage. As with any language, you must know the parameter passing, naming, and value return conventions. But there are scads of details that will trip you up.

First, thanks to the magic of type-safe linkage and the process of 'name mangling' (encoding class and type information into the function names), the names of C++ functions aren't nearly as simple as C or Pascal functions. Second, every member function in C++ has a secret parameter called `this`; you must know where `this` is and how to access the data members to which it points. Third, calling another C++ function from assembly can prove trying on the nerves as, once again, you must construct a mangled function name. And we haven't even touched the subject of virtual member functions (yet)!

A Short Review

In C linkage, all names are case sensitive (case is preserved), with an underscore (_) stuck in front. Parameters are pushed onto the stack starting with the rightmost parameter (the first parameter is at the lowest address). Stack space taken by the parameters is released by the calling program, after return from the function, by adding a value to the SP register.

Pascal linkage makes all names uppercase, with no added characters. Parameters are pushed from left to right (the opposite order of C). Unlike C, the Pascal functions clean up the stack as part of the return (using a `RET nn` instruction).

Both languages send back return values (of up to four bytes) in the AL, AX, or DX:AX registers. Return values of larger size are not standardised between compilers, and we will not discuss them here.

Most explanations of calling assembly from a high level language concentrate on what must be done with the BP register in order to access the parameters on the stack. MASM V5.1 removes all that drudge work. If you give a language qualifier in the `.MODEL` directive, MASM will not only convert the function name just as the Pascal or C compiler does, but will also allow you to list parameters and their types in the `PROC` statement, and use them in the function with the names you have given. And it will insert the proper 'prolog' and 'epilog' code in all functions to set up BP for use in parameter and local variable access. You'll never see BP anywhere in the functions, but be aware that it is in use, and must not be fooled with.

See Figure 1 for an example of an assembly function which can be assembled to be called directly from a Pascal or C program. If the language in `.MODEL` is C, an underscore is automatically added in front of all global labels (and all `PROC`s are made global). If the language is Pascal, nothing is added to the labels. MASM's `/mx` command line switch causes all global labels to be case sensitive (they are otherwise converted to uppercase). Thus, using the C qualifier along with `/mx` gives the name `_OutBytes` while Pascal without `/mx`, gives `OUTBYTES`.

Zortech C++ Linkage

While the Zortech compiler uses the 'standard' C linkage method for functions declared as C, for C++ functions it uses a hybrid of C and Pascal, with an extra twist.

C++ function names are given argument type information through a process called 'mangling'. Each name has an underscore

Type	*code*
char	c
short	s
int	i
long	l
float	f
double	d
long double	r
void	v
Prefixes	
near *	p
far *	P
reference	R
const	C
volatile	V
unsigned	U

Figure 2 - Mangling codes for simple data types

placed in front, and the case is preserved as in C. But tacked onto the end of the name (after two underscores) is an optional class name (for member functions), followed by type information for the function and its arguments. This is explained in detail in the Compiler Reference section titled 'Name Mangling in Zortech C++'. (an article on the same topic, but for Borland's Turbo C++, appeared in *.EXE Magazine*, February 1991) See the table in Figure 2 for a list of basic types and their encoding in a mangled function name. Classes and other complex types are encoded by giving a count of the letters in the class name followed by the name itself. In general, the mangled function name takes this form:

```
_<Name>__<n><Class><N|F><type-info>
```

where *n* is the length of the class name. *N* means 'near function', *F* 'far function'. So, for `Disp::Puts(char*s)`, we would have (in Small model):

```
_Puts__4DispNpc
```

read as '`Puts` is a member of class `Disp`, a `near` function with one argument, which is a `near` pointer to `char`'.

```
; INBYTES.ASM
; a non-member function with C++ Linkage
; input ct bytes of data from port to array at vals
;    masm /mx inbytes;


  .MODEL  SMALL, PASCAL ;near code, near data
  .CODE
  .186  ;enable assembly of 186 & up instr.
;;
;; void InBytes(int port, char *vals, int ct)
;;
_InBytes__Nipci PROC  USES DI, \
      port : WORD, vals : PTR BYTE, ct : WORD
  MOV    DX, port
IF (@datasize)
  LES    DI, vals  ; if far data model (C or L)
ELSE
  MOV    DI, vals  ; if near data model (S or M)
  MOV    AX, DS    ; all data is relative to DS
  MOV    ES,AX
ENDIF
  MOV    CX, ct
  CLD
IF (@Cpu AND 2)
  REP    INSB      ; only works on 186 & > CPUs
ELSE
IB_00:
  IN     AL,DX     ;simulate REP INSB
  STOSB
  LOOP   IB_00
ENDIF
  RET
_InBytes__Nipci ENDP
END


// INBYTES.H - prototype for INBYTES.ASM


void InBytes(int port, char *vals, int ct);
```

Figure 3 - A non-member function with C++ Linkage

They say the first Fortran 90 Compiler won't be available until at least 1992.

We say they're wrong.

NAG is pleased to announce the release of the world's first Fortran 90 Compiler – one of several exciting new products to be released in the NAGWare Service.

Fortran is still one of one of the most widely used scientific and engineering programming languages and the f90 Compiler will ensure it stays so. The NAGWare f90 Compiler is a full implementation of the ISO Fortran 90 Standard and is ready to meet the most demanding test of all – you.

The future is in the present.

NAGWARE™

For further information about NAGWare and the other products and services available from NAG, please contact:

NAG Ltd, Wilkinson House, Jordan Hill Road, OXFORD, OX2 8DR, UK, Tel: +44 865 511245 Fax: +44 865 310139 Telex: 83354 NAG UK G
NAG Inc, 1400 Opus Place, Suite 200, Downers Grove, IL 60515-5702, USA, Tel: +1 708 971 2337 Fax: +1 708 971 2706
NAG GmbH, Schleißheimerstr. 5, D-8046 Garching bei München, Deutschland, Tel: +49 89 3207395 Fax: +49 89 3207396

CIRCLE NO. 056

As more arguments are added to a function, their type information is appended to the end of the name. Return types are not included in the mangling information.

There are many other sneaky rules about name mangling. For example, if the type information of an argument is longer than two characters, and if another argument in the same function has the same type, it is encoded as *'Tn'*, where *n* is the position of the first occurrence of that type in the argument list.

The best way to learn the mangled name of a function is to write the function in C++ and compile it with debugging turned on (-g). Then examine the .MAP file produced by the linker. Alternately, compile the function, then run OBJTOASM on the output. Either method will show you the exact name, as mangled. Use this name as a model for your assembly function.

C++ Parameters

Surprisingly, in the Zortech C++ linking convention, parameters are passed as in Pascal - they are pushed onto the stack from left to right, and are cleaned from the stack by the `RET` statement of the called function. The reason for this is speed and compactness. The C method is more flexible, as it allows calling the same function with different numbers and types of arguments (eg `printf()`), but it has more instructions (the `RET` and stack adjustment are separate).

If a C++ function has an ellipsis in its argument list, C parameter passing is used instead. As these are infrequently used, we will ignore them for now.

In order to satisfy both the parameter passing and naming conventions, I use `.MODEL Pascal, and MASM /mx` to preserve case. As Pascal doesn't add the underscore (or the type information), I must construct the mangled function name myself. (At one time I had hopes of writing

Much of this information was obtained from hours of studying disassembled code with a debugger

a general purpose macro to mangle the name for me, but gave up when it proved easier to cut and paste from the MAP file.)

Using Parameters

Inside the assembly function, value parameters can be accessed by using the name given in the `PROC` statement. The only exception is `chars;` you can get to them with, eg, `byte ptr c`. This overcomes the fact that chars are declared as `WORD` in the parameter list to keep the stack aligned properly.

Pointers are just a special case of value parameters. The 'value' of the parameter is an address which points to the real data. To reach that data, you dereference the pointer. `InBytes and OutBytes` give good examples of that - load the address into a register and grab the data using register indirect addressing (for Small and Medium models):

```
MOV DI, vals
MOV AL, [DI]
```

or (for Compact and Large models):

```
LES DI, vals
MOV AL, ES:[DI]
```

References are really just pointers in disguise - an address that points to the data. In assembly, you can declare and use them just as if they were pointers. Arrays are also passed as a pointer to the first element of the array.

A Non-member C++ Function

Figure 3 shows an assembly function written to conform to the Zortech C++ linking convention. Because it isn't a member function, `InBytes` contains no class name in the mangled function name. The section of the mangled name following the double underscore tells us that `InBytes` is a `near` function taking an `int`, a `char*`, and another `int` as parameters. Notice that we had to add the leading underscore ourselves, as MASM only adds it when using C linkage.

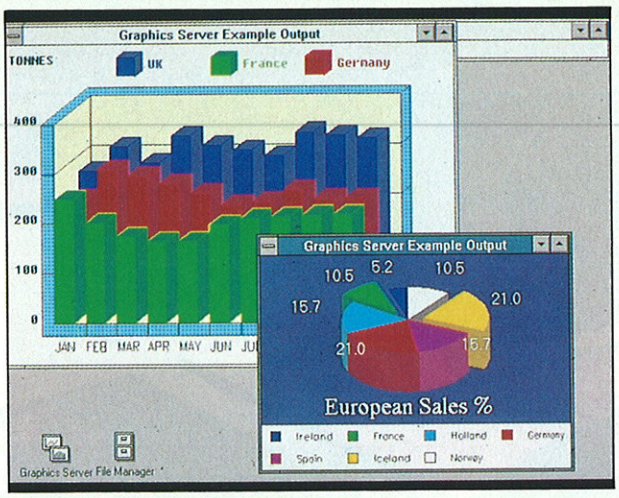
While `OutBytes` could be assembled for a different memory model with no change (except for giving the memory model in .MODEL), `InBytes` must have a new

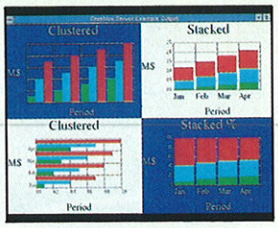
```
; DISP2.ASM - ASM member functions of class Disp
;     masm /mx disp2;


   .MODEL  SMALL, PASCAL


; asm equivalent of class Disp
Disp STRUC
  row    dw ?
  col    dw ?
  attrib dw ?
  height dw ?
  wid    dw ?
  base   dd ?
Disp ENDS


; for accessing the high and low words of a dd
Long STRUC
  Lo  dw ?
  Hi  dw ?
Long ENDS


   .CODE
;;
;; void Disp::At(int r, int c)
;;
_At__4DispNii PROC USES SI, r : WORD, c : WORD, thiss : PTR Disp
  MOV    SI, thiss       ; SI -> object of class Disp
  MOV    AX, r
  MOV    [SI].row, AX    ;save to object
  MOV    DH, AL          ;into reg for INT 10h
  MOV    AX, c
  MOV    [SI].col, AX
  MOV    DL, AL
  MOV    AH, 2           ;position cursor
  MOV    BH, 0           ;page 0
  INT    10h
  RET
_At__4DispNii ENDP
;;
;; void Disp::Put(char c)
;;
_Put__4DispNc PROC USES SI DI, c : WORD, thiss : PTR Disp
  MOV    SI, thiss       ; SI -> object of class Disp
  LES    DI, [SI].base   ; ES:[DI] -> screen
  MOV    AX, [SI].row    ;base + (row*wid) + col
  MUL    [SI].wid
  ADD    DI, AX
  ADD    DI, [SI].col
  SHL    DI, 1           ;adjust from byte ptr to int ptr
  MOV    AX, [SI].attrib
  MOV    AL, byte ptr c
  STOSW
  MOV    AX, [SI].col    ; set to next column
  INC    AX
  CMP    AX, [SI].wid    ; check for right margin
  JAE    DP_99
    MOV    [SI].col, AX
DP_99:
  RET
_Put__4DispNc ENDP


END
```

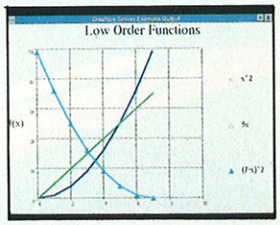
Figure 4 - Two member functions with C++ Linkage

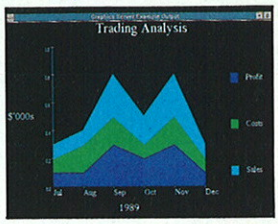
Bring Graphs and Charts to your Windows Applications Licence-Free with Graphics Server™

Introducing the **Graphics Server Software Development Kit (SDK)** from Bits Per Second. The most cost-effective, time-efficient way to integrate full-colour graphs and charts into your Windows applications. Here's why.









Client-Server Model

Graphics Server is a Windows server process which responds to multiple, concurrent requests for interactive graphing and charting functions from client applications. It enables graphs and charts to be incorporated into any Windows application having a Dynamic Link Library (DLL) or Dynamic Data Exchange (DDE) capability. This includes C programs, Superbase, SQL Windows, ToolBook, Word for Windows and many others. In fact, virtually all Windows applications!

Multiple Choice

Graphics Server lets you choose from 16 graph types, including 2-D and 3-D bar (simple, stacked and clustered), 2-D and 3-D pie, area, Gantt, scatter, polar, log-log/log-linear, line, high-low-close, X-Y and polar, plus a variety of statistical functions.

Ease of Programming

Graphics Server, through its multi-level interface, provides the level of control **you** want. In as few as **eight** calls you can open a new graph window, define titles, legends and labels, transfer data, draw a chart, and output the chart to the clipboard or printer. Graphics Server takes care of all the details; scaling the graph to match the data, matching fonts, text size and position to the graph, even resizing and repainting the graph window when necessary! You'll save development time. And money.

Alternatively, if you want full control yourself, Graphics Server provides over a **hundred** functions to help you design, create and manage virtually any kind of graphically drawn image. Multiple graphs, multiple windows, user interaction via hot regions. All these and more!

Full Support

The Graphics Server SDK gives you all the tools, documentation and examples you'll need to make immediate and effective use of Graphics Server. Within hours your applications will be transformed by the inclusion of presentation-quality graphics.

Telephone/fax support and a bulletin board service are provided for all registered users, and for power users requiring an even higher level of service and security there's a special VIP Support Programme.

No Licences!

The Graphics Server SDK costs £275 + VAT. No further licence fees to pay.

To place an order, request a demo disk, or obtain additional information, call . . .

BITS PER SECOND

Bits Per Second Ltd
14 Regent Hill
Brighton BN1 3ED
Tel (0273) 727119
Fax (0273) 731925

All product names are trademarks of their respective holders.

CIRCLE NO. 057

Desktop FORTRAN

EXPERTS AGREE WHAT TO BUY – THEY JUST CAN'T AGREE WHY!

"The . . . FTN77 compiler has allowed us to port our main-frame applications to 386 micros with minimum effort. The speed of compilation is incredible!. . . The run time performance is fantastic, full support for 32-bit operations and the 80387 coprocessor means our programs run like lightning!"
Tony Fitzpatrick, Exploration Consultants Ltd.

"I find the FTN77/386 compiler to be fast and easy to use . . . . We often download programs under development from a mainframe to the PC to run through this computer."
Rob McLaren
FTN77/386 User.

"Without the FTN77/386 compiler it would have been totally impractical for us to convert our million-line + power system analysis program (PSS/E) to run on a 386 PC."
Wayne B'Rells
Power Technologies Inc.

"FTN77/386 provides more intrinsic functions than any other: functions to manipulate strings, handle screen and keyboard I/O, read the date and time, parse the command line . . . Advanced file manipulation functions allow an FTN77/386 program to work with files in a C-like manner".
. . Scott Ladd, Computer Language, Nov. 1988.

". . . I am very impressed and have decided to shift all our production programs over to FTN77. . ."
Dr. G.R. Chaplain
NCL Investments Ltd.

"The ease of use, speed, debugging facilities, dynamic libraries with resulting small .EXE files, and extensive library routines made it a joy to use."
H. Gordon Jensen
FTN77/386 User.

"By far the fastest and most complete FORTRAN compiler that I have ever seen or used in the DOS environment".
Tim Wool
FTN77/386 User.

"The debugger was very easy to use. There is an on-line help facility which made reference to the manual unnecessary."
. . . Mike Gunn and Arul Britto, EXE Magazine, May 1989.

"For sheer speed and sleekness the (Salford) compiler stands out like a Ferrari among Yugos"
. . . Jack W. Crenshaw, Computer Language, May 1990

FAST, FRIENDLY & FULLY FEATURED – *THE PERFECT MIX – CALL US NOW!*

SALFORD SOFTWARE MARKETING LTD. MAXWELL BUILDING · THE CRESCENT · SALFORD M5 4WT.
TELEPHONE: 061 745 5678 · FAX: 061 745 5666

FTN77 386/486 – extending MS-DOS.

UNIX VERSION AVAILABLE NOW!

CIRCLE NO. 058

THE CONCEPT BEHIND OUR CASE PRODUCT.

System Architect is perfect for the beginning CASE user, yet has the power and flexibility to meet the needs of the most experienced users and largest applications.

System Architect has the power and flexibility you need from a CASE product. It works with today's most popular methodologies, including DeMarco/ Yourdon, SSADM, Gane & Sarson, Ward & Mellor (real-time), entity relation diagrams, decomposition diagrams, object oriented design (optional), state transition diagrams, and flow charts. It supports an integrated data dictionary/encyclopedia, and allows multi-user support both with and without a network. What's more, System Architect's open architecture lets you easily import and export data to other products.

Automated documentation facility

SQL-Like custom reporting

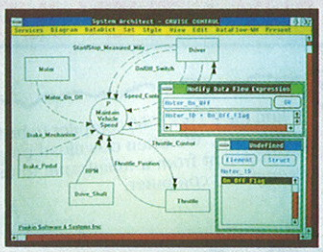
Windows based

Multiple methodologies

User-defined attributes

Requirements traceability

Rules & balancing



Multi-user/ Network version available

Matrix reporting

Auto levelling

Integrated data dictionary

Import/export capability

SSADM support

"System Architect stood out from many other prospects because it had the best CORE technology, as demonstrated by it's reliability, flexibility and performance."

Toshiba Corporation

Yet System Architect is more than just powerful. It's easy to use. Microsoft Windows-based and graphics oriented, System Architect lets you get up and running right away. And if you get lost, you can call on a context-sensitive help facility as well as a novice mode.

"It is very easy to learn and simple to use. We use System Architect at the heart of our project environment because it is so flexible in logical and physical terms of the data dictionary."

National and Provincial

At £1,110 System Architect is very affordable. It runs on any PC, and it won't run away with your budget.

"Think productivity has to be pricey? Think again. This product is truly a price performance leader."

System Builder

If you looking for a CASE product with power, ease of use, and affordability, look to System Architect. It's a concept whose time has come.

System Architect™

Supporting IBM's AD/Cycle

TO ORDER
TEL: (0926) 450858
FAX: (0926) 422165

MICROSOFT WINDOWS
Version 3.0 Compatible Product

Real Techniques & Methods Limited
118–120 Warwick Street, Royal Leamington Spa, Warwickshire, England CV32 4QY.

Please allow 14 days for delivery.

CIRCLE NO. 059

name when the model is changed. Medium or Large model will require the 'N' (near function) to be changed to 'F' (far function), while Compact or Large will necessitate changing the 'p' (near pointer) to 'P' (far pointer). This might seem like a nuisance, but it can save your life. Having the 'distance' of pointers encoded into the function name means that you will never accidentally link in a function compiled for the wrong memory model.

The clause 'USES DI' in the PROC statement tells MASM that it should PUSH DI on entry to InBytes, and POP DI before exit, preserving the original value. Never change the values of the SI, DI, BP, SP, DS, or SS registers without saving them first. Other registers can be changed without worry, as the compiler does not assume their values will be preserved across function calls.

Member Functions

Aside from the addition of the class name to the mangled function name, the only difference between a normal C++ function and a member function is the presence of the this pointer. this points to the object for which the member function was called; all accesses to the data members of an object in a member function are through this.

Zortech implements this as a hidden last parameter. Thus, a member function with the following prototype:

```
void Disp::Put(char c);
```

could be thought of (in terms of parameter passing, **not** naming) as:

```
void Put(char c, Disp *this);
```

(The Zortech V3.0 manual incorrectly states that this is a hidden first parameter. Believe me, it is the *last* parameter.)

To access the data at this, add another parameter to the end of the PROC declaration:

```
_Put__4DispNc PROC \
        USES SI DI,\
        c : WORD,  \
        this : PTR Disp
```

(note that I declared c as a WORD rather than BYTE, because char parameters are passed as a word to retain word alignment on the stack.) The above *almost* works. The only problem is that, in MASM, this is a reserved word. In my programs, I always call it 'thiss' instead.

Data Members

'thiss' points to the first data member of the object. To use the other data members, you must make a STRUC with the same fields as the data fields of the class. You can then access the data by loading the pointer in thiss into an address register and using the fields of the STRUC as offsets from that address. For example, to get to the height member of Disp (Figure 4), use:

```
MOV  SI, thiss
MOV  AX, [SI].height
```

in Small or Medium model, and

```
LES  SI, thiss
MOV  AX, ES:[SI].height
```

in Compact or Large model.

```
// DISP.H - definition of class Disp

typedef unsigned char BYTE;
typedef unsigned int  WORD;

class Disp
  {
protected:
  int row, col;           // current row and column
  WORD attrib;            // colour to use when outputting
  int height, wid;        // size of display
  WORD far *base;         // ptr to 1st char of display
public:
  Disp();                 // open display
  void At(int r, int c);  // position cursor at row=r, col=c
  void In(BYTE f, BYTE b) // set colour for writing
    { attrib = (f << 8) | (b << 12); }
  void Put(char c);       // put c at current position
   void Puts(char *s);    // put s at current position
  };  // class Disp

// end of DISP.H

// DISP.CPP - members of Disp implemented in C++

#include  <dos.h>    // for REGS, int86, MK_FP
#include  "disp.h"

Disp::Disp()
  {
  REGS reg;

  // learn the current video mode and width
  reg.h.ah = 15;
  int86(0x10, &reg, &reg);
  base = (WORD far *) MK_FP( ((reg.h.al == 7) ? 0xB000 : 0xB800),
0 );
  wid = reg.h.ah;

  // EGA and above store (height-1) at 40:84h
  height = *((BYTE far *) MK_FP(0x40,0x84))+1;
  if (height < 25) height = 25;

  // find initial cursor position
  reg.h.ah = 3; reg.h.bh = 0;
  int86(0x10, &reg, &reg);
  row = reg.h.dh;
  col = reg.h.dl;
  In(7,0);
  }  // Disp::Disp()

void Disp::Puts(char *s)
  {
  char ch;
  while (ch = *s++)  // get characters until 0
    Put(ch);
  }  // Disp::Puts()

#if defined(NOASM)
// C++ versions of assembly functions
void Disp::At(int r, int c)
  {  // position the cursor to row r, col c
  row = r; col = c;
  REGS reg;
  reg.h.ah = 2;
  reg.h.bh = 0;
  reg.x.dx = (row << 8) + col;
  int86(0x10, &reg, &reg);
  }  // Disp::At()

void Disp::Put(char c)
  {  //  put the character c at current position
  WORD far *pos = base + (row*wid) + col;
  *pos = attrib | ((BYTE) c);
  if (++col >= wid)
    col--;
  }  //  Disp::Put()
#endif // defined(NOASM)

// end of DISP.CPP

// DISPTEST.CPP - test the Disp class
//
//    ztc disptest disp disp2    - to test with ASM functions
//    ztc -DNOASM disptest disp  - to test w/o ASM functions
//
//    note that ztc will call MASM /mx for given ASM files
//
#include  "disp.h"

int main()
{
  Disp mono;
  mono.At(20,5);
  mono.Put('X'); mono.Put('Y');
  mono.At(4,15); mono.In(0, 7);
  mono.Puts("This is an entire string.");
  return 0;
}
// end of DISPTEST.CPP
```

Figure 5 - DISP.H, DISP.CPP, and DISPTEST.CPP

Be careful when constructing the STRUC that you match the layout of the class exactly. Pay attention to the size of each element (ints are 'DW', longs are 'DD', data and code pointers are 'DW' or 'DD' depending on memory model). Remember that, unless you use the -a compile switch (or the align #pragma), all data members except char are aligned to a word (even byte) boundary. The safest way to determine the offset of each data member in the class is to write a program which subtracts the address of the start of an object from the address of each its members and displays the results. For example, to find the offset within the class of Disp::col, use:

```
Disp t;
printf("%d",(long)&t.col
              - (long)&t);
```

Don't forget that some variables change size with different memory models.

Figures 4 and 5 give a complete example which implements the Disp class. Disp implements basic positioning and output functions for the PC display, taking into account the screen dimensions reported by the ROM BIOS. Two of the member functions are implemented in assembly (At, Put), while the others are in C++. Writing just these two short functions in ASM saved 81 bytes, according to the MAP file and my arithmetic.

You will get two warnings when you compile this example. Don't let either one bother you. One is warning about an assignment inside a while statement, the other about type-casting a WORD parameter to BYTE.

Calling Member Functions

After understanding all the above, calling a member function from assembly should seem trivial. Just push the arguments to the stack, left to right, with a final push of thiss just before calling the function which, by the way, has a funny mangled name. To call Disp::At(4,10) (from within another member function of Disp), you would use:

```
PUSH  4  ;assumes 286
PUSH  10
PUSH  thiss
CALL  _At__4DispNii
```

With memory models other than Small, you need to adjust some things accordingly. In particular, Compact and Large models require that you push both the offset and the segment of thiss:

```
PUSH  thiss.Hi
PUSH  thiss.Lo
```

Virtual Member Functions

If the function you are writing will be called as a virtual member function, you need change nothing; the address of your function will be entered in the virtual tables automatically. If you want to call a virtual member function from your assembly function, it's not so easy.

Briefly, the addresses of all virtual functions are kept in a table which is stored at the address given by the final word (Small, Medium) or dword (Compact, Large) of the object data. The virtual function pointer table consists of word (Small, Compact) or dword (Medium, Large) pointers to the proper functions for this object. The first word (or dword) of this table seems to always be 0, while the following entries are the addresses of the virtual functions for this particular object, in the order they are encountered in the class declaration (see Figure 6). For example, if Disp::Put was a virtual function (and the first virtual function of Disp), you could call it in Small model (after pushing the argument and thiss) with:

```
MOV  SI, thiss
MOV  BX, [SI+size Disp]
CALL [BX+2]
```

The second virtual function would be called as [BX+4] (again, for Small model), and so on.

If you really want to get involved in something so scary, check it out with a debugger first. My advice is to avoid it unless absolutely necessary, and if you really do need to call virtual members from assembly, don't believe anything you read in a magazine - research it for yourself.

Be careful that you don't call virtual functions directly from assembly - call them indirectly through the virtual function table if you must call them at all. Direct calling may work as long as all objects are of the class you have defined. But just let someone inherit your class into a new class and redefine the virtual function. Pow! the new class doesn't work because the members of your base class are hard-coded to call the base class function, rather than calling through the virtual function table.

Dangers

Some practices are almost guaranteed to cause problems. The most obvious is calling inline member functions. As these functions don't really exist anywhere, you can't call them from assembly.

It is also probably best to avoid using assembly for functions that return any structs or classes, as the methods by which this is done are not standardised.

Finally, constructors and destructors contain lots of magic code. Never attempt to write a constructor or destructor in assembly!

Using Turbo C++

Turbo C++ brings even more complications to the writing of member functions in assembly. First, the names are mangled differently (as outlined in *.EXE February 1991*). Just to make you sweat, Borland's utilities unmangle the names before putting them in the MAP file, requiring you to either search through a hex dump of the OBJ, or construct the names yourself, according to the rules.

Second, although Turbo C++ uses C parameter passing (push the rightmost parameter first, caller cleans the stack), the mangled names have no leading underscore. The lack of underscore prevents us from using .MODEL C, but when we use .MODEL Pascal, RET instructions are always assembled to clean the stack!

My suggested solution to this mess is:

- Use .MODEL Pascal.
- Construct the name by hand, or write a program to do it.
- List the parameters in reverse order (with thiss at the end).
- Instead of RET, use RET 0 to force vanilla RET instructions (which don't clean the stack).
- If you call a C++ function from within your assembly code, don't forget to clean the parameters from the stack when you're done (with, eg, ADD SP, 2).

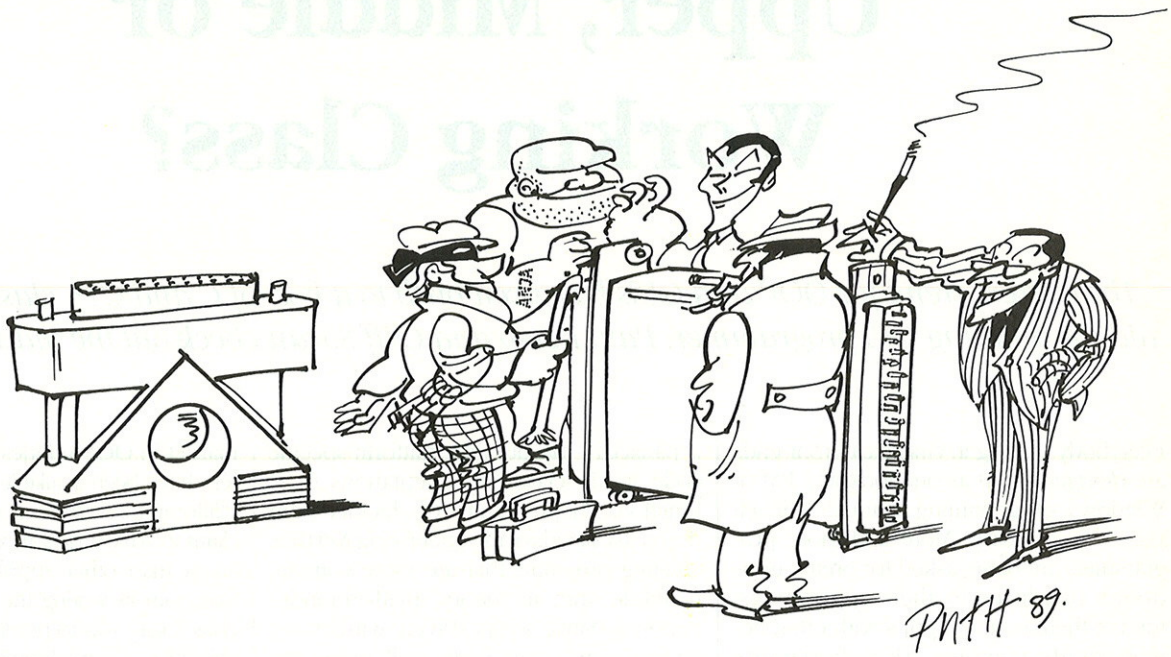
Conclusion

Although fraught with perils, writing member functions of C++ classes in assembly language is possible. In return for the extra work, you get smaller, faster programs, and a better understanding of how your compiler works.

EXE

Laine Stump was born near Yellowstone Park in Cody, Wyoming (the North-western US) and has been living in Turkey for the last six years. He currently teaches assembly language and C++ courses at Bilkent University in Ankara, Turkey, where he lives with his wife, Ayfer, and cat, Mahmut.

5 out of 5 hackers prefer other software protection methods to Hardlock E-Y-E.®

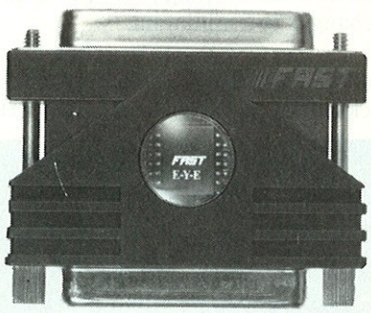


What hackers dislike...
Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's No. 1 in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool. Hardlock E-Y-E is based on a custom chip featuring secure algorithmic response rather than simple bit swapping or counting schemes.

What software developers like...
Hardlock E-Y-E combines all the features software developers require in a single product: algorithmic response to provide security and an optional non-volatile memory to allow custom configurations. FAST Electronic has made implementation of Hardlock E-Y-E in your software easy. Use HL-Crypt to protect .EXE or .COM files, or incorporate high level language interface routines in your software. The algorithm parameters and the contents of the memory can be programmed in seconds using our Crypto-Programmer card. This unique card guarantees that no one else can burn your original codes. Simply plug the card into any PC slot and start up your own Hardlock E-Y-E workshop.

What your customers will like...
Hardlock E-Y-E allows unlimited backup copies. The device is shipped with the software for the user simply to plug into the parallel interface and forget. Daisy chainability, outstanding reliability (no battery is needed), and the most compact High-Tech design ensure that your customer will accept Hardlock E-Y-E.

What your accountant will like...
Hardlock E-Y-E needs no factory coding. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piracy and multiple usage are prevented. Despite its wealth of features, Hardlock E-Y-E's prices remain competitive.



Hardlock E-Y-E programmable, algorithmic response and memory option – all in one.

...As more and more software developers, customers and accountants appreciate the Hardlock E-Y-E device, hackers like it less and less.

MAGNIFEYE

Order your demo unit today. Contact Magnifeye, 235-239 Walmer Road, Walmer Studio #6, W11 4EY, Telephone 071 221 8024, Fax 071 792 3449.

CIRCLE NO. 060

Upper, Middle or Working Class?

The current trend for OOP and GUIs has given birth to a host of C and C++ class libraries to aid the budding GUI programmer. Paul Kemp and Cliff Saran check out the main contenders.

Everybody's doing it. Graphical front-ends are *de rigeur* - but as one look at a PM or Windows programming manual will tell you - it's not easy. More and more programmers are being asked for pretty, user-driven interfaces to their applications, forcing them to get to grips with often bewilderingly complex APIs. Portability becomes a major problem as a larger proportion of an application's code becomes 'Presentation System' specific. Developers that have invested a great deal of programmer time and money into creating applications with a fancy GUI understandably want their products to be available on as many platforms as possible, while keeping port-related coding changes to a minimum.

The adoption of object-oriented techniques to tackle these problems would seem to be a panacea - by burying platform-specific code inside class implementations or C functions, the programmer is shielded from the messy details of the underlying API thus creating programs that are more concise, readable and, in theory, platform-independent. Since event-driven windowing environments map fairly well on to the OOP model (with messages passing between window 'objects'), toolkit vendors have been beavering away and released a number of C++ class libraries and 'object-oriented' C libraries to ease the pain of GUI programming. In this article we survey some of the current offerings.

CommonView

Veteran Hibernian OOPer John Carolan and his merry band of object wizards at Glockenspiel were the first outfit to exploit the nascent market in OOP utilities. Their C++ compilers have been available on a wide range of platforms for some time, so they can justifiably claim to have a deal more experience in this arena than other suppliers. Glockenspiel's main aim in writing the CommonView C++ class library was to provide a genuinely portable object-oriented application framework. Its commitment to this rigorous design criterion is reflected in the somewhat limited range and functionality of classes in the CommonView hierarchy. For example there is no explicit Clipboard, Timer or DDE support because such facilities are too platform-dependent to warrant inclusion in the library. This approach appears to have paid off however and at the moment CommonView is available on a number of platforms: Windows, OS/2 PM and OSF/Motif with HP support in the pipeline.

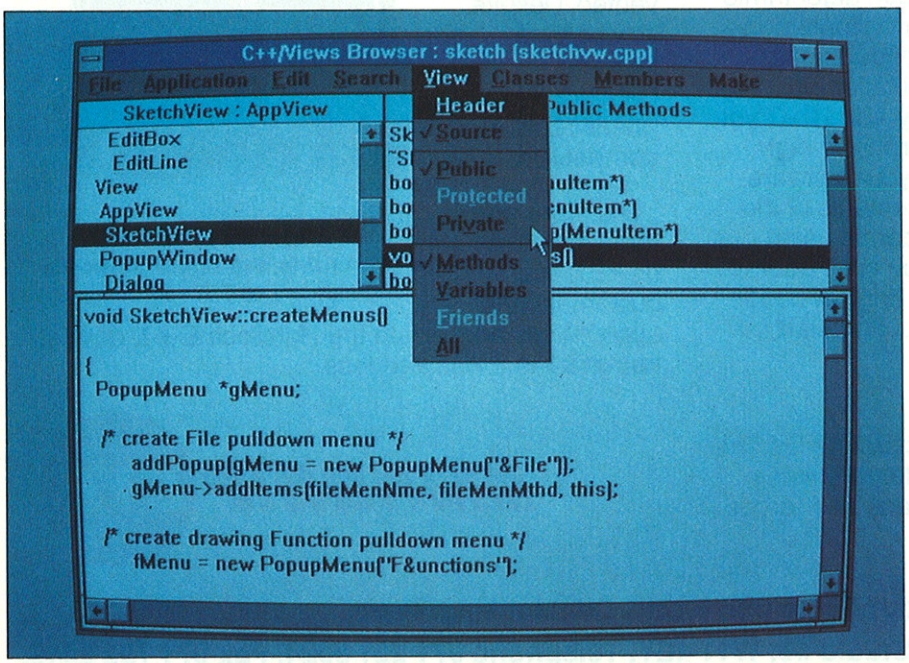


Figure 1 - The C++/Views Class Browser

The CommonView package actually consists of three main component hierarchies: FreeStore (memory allocation), Container (arrays, tables, lists etc.) and CommonView itself - the GUI library. The latter comprises some 60 standard GUI object classes such as `DialogWindow`, `Control` and `Menu`. Event handlers, in the form of virtual member functions of the `Window` class, are provided to trap the most common message types in an event-driven system (eg `Expose()`, `ReSize()` etc) which simply need to be re-implemented if your application wishes to process these messages. If you need to handle messages like `WM_TIMER`, for which there is no event handler, it is necessary to override the `Dispatch()` method of `Window` and interrogate the raw `Event` object received to determine the actual message ID and associated parameters.

Extending the functionality of the CommonView hierarchy by writing platform-

specific code (which you will almost certainly have to do at some point) is achieved by obtaining hooks into the underlying presentation system through invocation of the `Handle()` method of classes such as `Window` and `Menu`. This returns a generic CommonView `HANDLE` which must be cast to the appropriate data type (such as an `HWND` in PM) for use in an API call. 'Kickdowns', as they are known, should of course be avoided if possible, as they compromise the portability of your code.

CommonView is a robust, no-frills framework which, because of its ascetic modular design, should be (if you're careful) portable and *interoperable* - ie easy to integrate with other third party libraries. Nearly all the classes are implemented in DLLs, so porting and upgrading is simple and the executables are small and efficient.

CommonView for Windows 3.0 costs £395, requires the Borland, Zortech or Glock C++ compiler and the Windows SDK (except if you're using Borland's C++ 2.0 for Windows). OS/2 PM version also costs £395 and supports the Zortech or Glock compiler and requires the PM SDK. OSF/Motif versions range in price from £1,470 to £9,450 depending on the platform. Contact QA Training Tel: 0285 655888 or Glockenspiel in Dublin on 010 353 1 733166.

C++/Views

Unlike CommonView, C++/Views from CNS Inc is a cosmic GUI class library. This means that all of the 75 or so classes provided in the hierarchy are derived from an ancestral generic `Object`. C++/Views also differs from CommonView in that it is bundled with a number of utilities, and on initial inspection seems to provide a greater range and functionality of objects. Both foundation and framework classes are built into the library, including an `Archiver` class which implements the persistence of objects by providing a mechanism for object storage and retrieval on disk. The product is mainly aimed at improving the productivity of Windows 3.0 developers and only supports that platform at present (PM, Motif and Mac variants are purportedly on their way before the year's end).

C++/Views implements what is known as the 'Model-View-Controller paradigm' or MVC (CNS's words, not mine). The general idea here is that the separation of an application into these three interchangeable layers provides for greater portability - for instance you just plug in a different View layer for each presentation system (isn't that what OOP is all about anyway ?). The *Model* layer is where application-specific

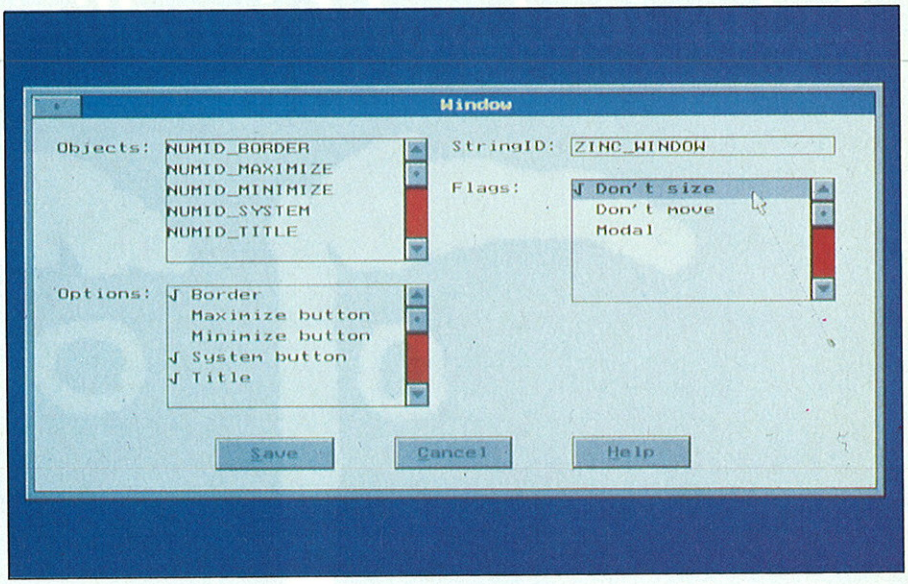


Figure 2 - Interactive screen design using Zinc's Windows Editor

classes sit. They are usually written by the development team and may use foundation classes and/or add-on classes for tasks such as database access. The *View* layer implements the graphical elements of the application (hence C++/Views geddit ?). These objects extract information from the model layer and present it to the user (in graphs, lists etc). At the top sits the *Controller*. Objects in this layer supply the interface between input devices (eg mouse, keyboard, timer) and the model and view layers. The controller layer of C++/Views consists of only one object class called the `Notifier`. It has application scope, is instantiated only once and serves to initialise the windowing environment for event processing (`Notifier`'s `start()` method is analogous to the message polling loop in Windows or `App::Exec()` in CommonView).

There are a couple of handy development tools included in the C++/Views package: a Smalltalkish Class Browser and a Dialog Code Generator. The browser is a Windows program that can be used to maintain and document the classes used in an application (see Figure 1). The Dialog Code Generator is an interesting program which reads a .DLG file created with the SDK dialog editor and generates a skeleton C++ source file to implement the design.

All in all, C++/Views provides a fairly comprehensive and extensible framework for Windows development. However, because the library is not supplied as a DLL, upgrading and porting will require at least relinking and executables don't come any smaller than about 100K. Another worry is that because of its cosmic inheritance tree it may prove difficult to use with other cosmic libraries.

C++/Views for Windows 3.0 is supplied with complete source code, requires the Windows SDK and costs £200 + VAT if you're using the Zortech C++ compiler. Zortech distributes its version in the UK and are on 081 3167777. For prices of C++/Views which support the Glockenspiel and Borland compilers contact CNS on 010 1 612 9440170.

TIER

The TIER C++ class library from Genesis Development Corp was written by one Carl Sturmer and was not initially designed to be marketed as a product. It was used as an internal development tool by the Sturmer Hauss Corporation to aid the creation of commercial GUI applications before they decided to tart it up and flog it as an application framework. By the company's own admission it has, up to now, been sold pretty much by word of mouth and as yet there is no UK distributor.

Like C++/Views, TIER has a layered or *tiered* architecture: *TIER 1* is the C++ interface to the base platform primitives (this is currently available for Windows 3.0); *TIER 2* consists of a more generic set of higher level 'helper' classes and *TIER 3* is a totally generic, platform independent programming environment. While conceding that portability is important, Genesis also recognises that programmers who are writing serious Windows applications need access to the base platform functions. Consequently, code written using TIER 1 classes has a degree of platform dependence (since this layer will look slightly different for each target GUI).

The hierarchy is not cosmic but most of the classes are derived from `TlObject`, so it can be considered something of a half-

Just Released Object Graphics for Turbo Pascal

Actor just got friendlier...

Actor is an interactive development language for both new and advanced Microsott Windows programmers. Relying on the most modern object-oriented programming techniques. Actor boils down the complexitities of the Windows API to a few powerful principles called ObjectWindows.

Actor delivers Object - Windows in an easy-to-use programming environment ideal for learning Windows while writing your programs one step at a time. Add to this support for C libraries and DDE, and you've got the fastest route to Windows. Order your copy today.

Actor 3.1 Now Only £175

(includes ObjectWindows)

Actor Professional £375

(includes ObjectGraphics & The Whitewater Resource Toolkit)

Pure Object Oriented Programming for Microsoft Windows

For more information, upgrade requests or to place your order call NEOW Ltd on 0628 668334.

NEOW Ltd. Unit 12, Progress Business Centre, Whittle Parkway, Slough, SL1 6DQ. Tel 0628 668334. Fax 0628 665722.

CIRCLE NO. 061

way-house. The design of the classes themselves also appears to be something of a C/C++ hybrid. There are no event handlers or interaction objects (as in CommonView) and most of the methods look like Windows/PM API calls. This approach is deliberate, as TIER was designed to allow access to low-level functions and not 'fight' with the Windows C API. It does mean however that it doesn't use as many object-oriented features as other libraries and I suspect that there are not many benefits to be had from code reduction over a standard Windows app, although maintainability would be enhanced.

Because it is not doggedly committed to platform-independence, TIER 1 provides a very impressive array of 115 GUI classes. All Windows 3.0 functionality is encapsulated including the full complement of GDI graphics functions, so there should be no need for kickdowns. My feeling is that TIER might be a good library to choose if you are already a pretty competent Windows programmer wishing to move to C++ without losing any control over exactly what is going on below the OOP bedsheets.

TIER 1 for Windows 3.0 is currently on special offer at $199 + shipping (Air Mail $20, Courier $66) and includes support for Glock, Borland and Zortech compilers. The Windows SDK and a C++ compiler are required. Source listing costs $495. It is available direct from Genesis, 1303 Columbia Dr. Suite #209, Richardson, Texas 75081. Tel: 010 1 214 6448559 or fax: 010 1 214 6444286.

Astronomical Jargon

Foundation Library - a set of classes that implement basic programming objects like strings, arrays and streams. Abstract classes may also be supplied that define common behaviour when copying and comparing objects derived from the same parent.

Application Framework - a class hierarchy that is concerned with building an application's user interface (like the ones discussed in this article). Typical objects include windows, menus, buttons and some kind of event handling system.

Add-on Classes - eg. third-party libraries to support things like databases, specialised graphics, comms etc.

Application Classes - these are designed and implemented by the developer and are specific to the application being written.

Cosmic vs Non-cosmic - CommonView is described as a *non-cosmic* class library. This means that it is made up of independent component hierarchies, each one concerned with a specific task: memory allocation, user interface management, object storage etc. Libraries such as C++/Views and Win++ are *cosmic* because all objects are derived in Smalltalk fashion from a single parent class (usually called something like 'Object'). The `Object` superclass provides generalised methods for such things as comparing, cloning and archiving objects which are its descendants (subclasses). This would seem like a good idea except that, as Glockenspiel's John Carolan points out, there are a number of problems:

In the absence of an industry-standard set of foundation classes, some third-party vendors are supplying their own implementations of fundamental objects such as strings, streams and arrays. This can lead to chaos with name-clashes and parameter incompatibility when one tries to use two or more cosmic libraries within the same application (needless to say Glock's CommonView does not suffer from this problem and has, we're told, been tested successfully with C++/Views, RogueWave and Microsoft's AFX).

Secondly there is an overhead in terms of the amount of static data within an application, since *all* classes inherit `Object`'s data and methods. Developers can also come unstuck if the vendor who supplied their cosmic class decides to change its implementation or design since all their application classes are dependent.

Finally, the use of a cosmic superclass affects the link characteristics of a program so that (when using a standard linker) all virtual functions appear in the executable, whether they are called or not, resulting in bulkier .EXE files. Perhaps with the advent of Microsoft C++ and its attendant cosmic foundation library a *de facto* standard will emerge to dispense with many of these universal problems.

GUI_Master

Dutch software house Vleermuis is the originator of GUI_Master, a sophisticated object-oriented toolkit for developing Windows and PM applications (OSF/Motif should be available in December '91). The product is composed of three main parts: The Interface Builder, The Browser and The Vleermuis C++ Class Tree.

The Interface Builder is essentially a screen painter++. It lets you construct windows and dialog boxes on the screen and then generates C++ source code and resource specifications to implement the design. The Class Browser is another Smalltalk-like application similar in appearance to the C++/Views browser. It allows you to locate methods and classes and associate them with the source files that implement their behaviour. From here you can also launch your favourite editor to modify the code if necessary. The class tree is cosmic and pretty extensive, with 85 objects which implement the GUI plus some extra goodies that augment the functionality of the basic Windows/PM SDK. These include facilities for data scrolling in windows; undo-redo; clipboard support; pre-defined standard dialogs (such as 'Open', 'Save' and 'Font'); printing; archiving (persistence) and run-time error handling.

Vleermuis has big plans for the future of its C++ product family with several new libraries/toolkits currently under development. The first, DRAW_Master (a generic drawing program for the development of graphics applications), is in beta test at the moment and scheduled for release in September of this year. Other utilities in the pipeline are DATA_Master (an object-oriented shell for SQL databases); MATRIX_Master (for spreadsheet apps); QUIZ_Master (*sic* - for rule-based apps); AD/CYCLE_Master (to help build CASE tools that interface with IBM's repository) and COMS_Master (which implements mixed-vendor program to program communication). All these products will be shipped in source code format. GUI_Master looks like a well integrated GUI development environment but, as with a number of other libraries we have looked at, once you start down the Vleermuis track you might find it difficult to change course when new horizons beckon.

GUI_Master requires the Windows/PM SDK and a C++ compiler - Glock, Zortech or Borland (Windows only). Single user

licence for PM costs $495 and $545 for Windows. Site licence (combined PM and Windows) costs $4,950. Full non-competitive source code licence is priced at $50,000. Vleermuis is based in Utrecht on 010 31 30324944.

Zinc Interface Library

The Zinc Interface Library (ZIL) is a C++ class library that allows the applications programmer to create a professional GUI without having to resort to using Microsoft Windows. Both Windows and DOS based versions of the library are available. In fact, only minor modification of the source code is required to convert an existing DOS application to Windows. With C++ finding a new niche with developers of Windows applications, it's nice to see a GUI class library that supports both platforms. There are separate versions of the library providing support for both Borland's BGI graphics routines and Zortech's Flash Graphics routines. For those of us who can't afford trendy graphics displays, there is even a text-only mode of operation.

Conceptually, the ZIL can be divided into three main sections. The event queue provides the backbone of the system, with all events and interrupts being governed by the Event Manager class. User input is processed through Zinc's event queue before being passed onto the Window Manager class. This, in turn, controls windows and window objects by monitoring their priorities and screen positions. Operations like cutting, pasting and copying are handled automatically by the Window Manager. The final section contains all of those special features that complete the picture; there are classes for 'help' and error handling and for mapping of the palette. The Event Manager is capable of determining the attached display type automatically, and it is even possible to switch between a text-only and a graphical user environment at run-time. Consequently, an application can be written generally, without regard to any given display adapter.

To simplify the creation of application screens, the ZIL is packaged with the Zinc Designer (see Figure 2). This is an interactive design tool that enables screen layouts to be saved to disk as objects. These may be later retrieved using a single line of code.

The DOS version of the Zinc Interface Library costs $199.95 and $249.95 with source code. The Windows version also includes the DOS library and costs $299.95 and $399.95 with source code. ZIL is distributed in the UK by Zortech on 081 3167777. For more information contact Zinc Software Incorporated, 405 South 100 East 2nd Floor, Pleasant Grove, Utah UT 84052, tel 0101 801 7858900.

TEGL WINDOWS TOOLKIT II

TEGL Systems Corporation also ships a DOS base GUI library. At present, only C and Turbo Pascal versions of the library are available, but TEGL is working on an object-oriented version, which is expected to be released in August and this will be available as a free upgrade. The TEGL WINDOWS TOOLKIT II is supplied with a GUI library, a memory manager and a windows manager. There is no need to purchase a separate windowing environment or to use third party memory management software.

A GUI class library is just providing an object-oriented interface to the API beneath

The TGI graphics library is a go-faster version of Borland's BGI library. TEGL claims that its library runs between two to three times as fast as Borland's BGI. This is not surprising since the TGI is written in assembler and accesses video hardware directly. The TGI contains video drivers for CGA, EGA, VGA and Hercules graphics, and it also supports SVGA. The library may be used completely independently of the memory manager or windows manager.

As with Zinc's offering, TEGL's windows manager is event driven. Only a drawing event for a specific window object needs to be coded, as resizing of windows is done automatically. It is entirely possible to create an application based on a completely customised GUI and all the main windowing structures have been built into the library. Scroll bars, window borders and headers have already been defined, providing a rapid method of generating user friendly front ends for applications.

The TEGL Windows Toolkit II is supplied with an icon editor, more than 40 bit-mapped fonts and a font editor that can be used to create fonts as large as 100 x 100 pixels. There are versions of the library available for Turbo Pascal and C, with the C version supporting Turbo C/C++, Quick C, Microsoft C and Watcom C.

The complete TEGL Windows II package costs $99 and includes the complete source code. For more information contact TEGL SYSTEMS CORPORATION, Suite 780-789, West Pender Street, Vancouver, British Columbia, Canada V6C 1H2, tel 0101 604 6692577.

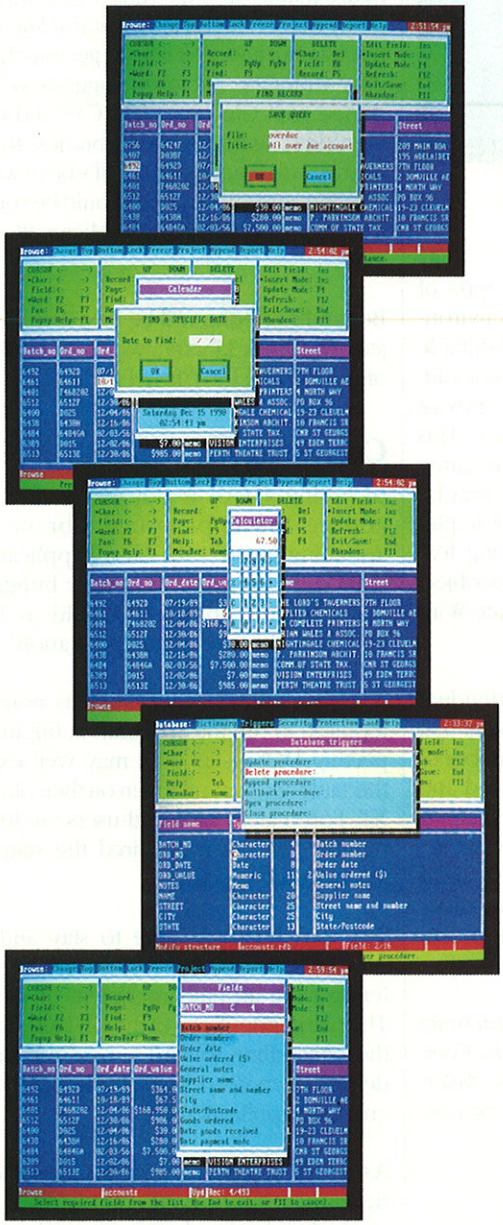
XVT

The Extensible Virtual ToolKit (XVT) is a GUI library that provides a portable interface for programmers who are developing applications intended for a multitude of platforms. At present the library is available for OSF/Motif, OPEN LOOK for X Windows, Microsoft Windows, Presentation Manager and the Macintosh (see Figure 3). There is also a text-only windows manager. Identical source code runs on all the above platforms, illustrating the versatility and the rugged design of the XVT library.

There are two versions of the library available; the C version (XVT) is totally compatible with ANSI C, and the C++ version (XVT++) supports AT&T's C++ V2.0. XVT is also planning on supporting Borland's C++ in a future release of the library. Superficially, it would seem that an object-oriented version of the library is actually available, but a closer look reveals that XVT++ is in fact providing an object-oriented front-end to the inner workings of the XVT interface underneath. There is a school of thought that states that a class library should be designed from the bottom up, adhering strictly to the object-oriented philosophy of programming in C++. On the other hand, a GUI class library is just providing an object-oriented interface to the API beneath. The XVT++ library is simply providing an additional layer of interface code to the API and as there is little performance penalty, using XVT++ to provide an object-oriented interface shouldn't really cause any headaches.

XVT does not include a design tool with its library but an interactive design tool and application generator is available as a separate package. XVT-Design is capable of generating GUI resources such as dialog boxes, windows, controls and menus which are portable and can be easily transferred across different platforms. It can also generate the C source for a framework application, including header files and the MAKE file.

Recital 7.0 Is The Only Choice.



- Advanced RDBMS/4GL
- Active Data Dictionary
- Object-oriented with pre-defined "application objects"
- Open client/server architecture
- Integrated pop-ups for calendar, calculator, choice lists
- Built in "Dynamic Compiler" New v7.0
- Pop-up time management scheduler New v7.0
- Relational multi-record table fields in forms New v7.0
- Split screen relational browse New v7.0
- Database, form and report triggers New v7.0
- Automatic record locking
- Supports 14 datatypes including memo, numeric, byte, integer, short, character, date, logical, float, real, packed, zoned, vaxdate, quadword
- Available now on 100+ VAX/VMS and UNIX computers (including full VAX cluster support)
- SQL (interactive and embedded)
- dBASE, FoxBase and Clipper compatibility (yes, even on VMS!)
- Terminal (including full color support) and keyboard independence
- Transparent connectivity to existing RMS and RDB databases on VAX/VMS
- Completely integrated development environment
- Powerful end-user tools

Recital CORPORATION

Rdb SVP SOLUTIONS • VENDOR • PROGRAM

88open COMPATIBILITY CERTIFIED

Recital Corporation LTD, South Bank Technopark, 90 London Road, London SE1 6LN, United Kingdom, Tel 071-401-2727, Fax 071-633-9617

Recital Corporation Inc., 85 Constitution Lane, Danvers, MA 01923, Tel (508) 750-1066, Fax (508) 750-8097

CIRCLE NO. 062

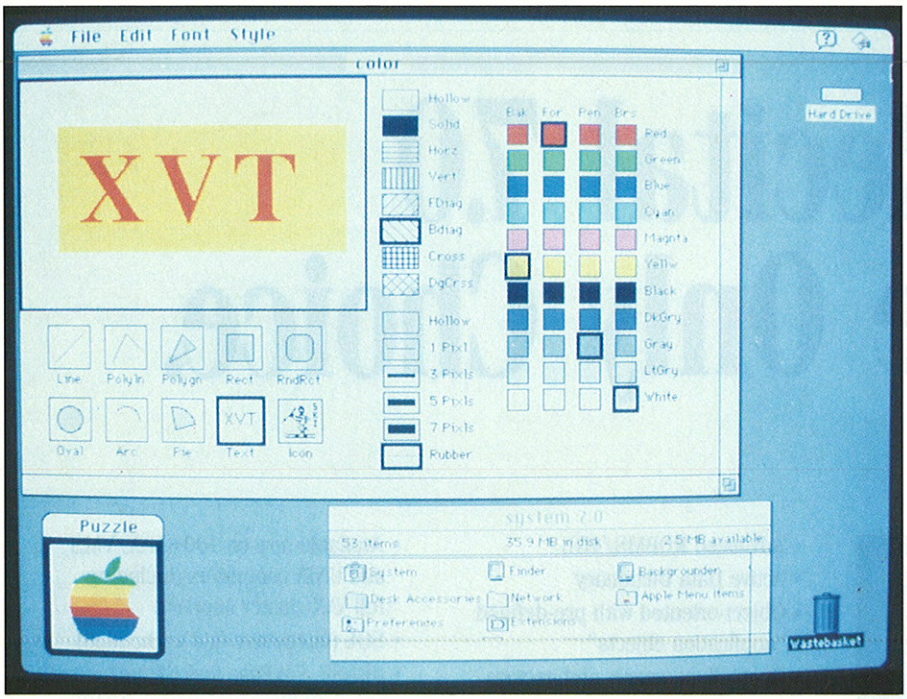


Figure 3 - An application using XVT on the Apple Macintosh

The XVT library for the Macintosh, OS/2 Presentation Manager or Microsoft Windows costs £545 and £3795 with source code. The X Windows version costs £995. Developers wishing to use a C++ GUI class library may purchase, for an additional £195, the XVT++ object-oriented interface to XVT, which will work on any of the supported platforms.

XVT-Design costs £595 for the Macintosh, Presentation Manager and Windows, and £1595 for the X Windows version. XVT is distributed in the UK by Personal Workstations Limited, 48-50 Gainsford Street, Bachelor's Wharf, London SE1 2NE and can be reached on 071 4036698.

Win++

Win++ is a C++ GUI class library that provides developers with a ready made environment for creating applications running under Microsoft Windows. Presently, Win++ only supports the Borland C++ V2.0 compiler. Most of the classes provide a straightforward, high-level interface to Windows. There are classes for all windowing objects and there are also container classes for handling such abstract objects as stacks and queues. Win++ is distributed in a number of formats including static-link and dynamic-link libraries. There is no need to purchase the Microsoft Windows Software Development Kit although it may be helpful, especially if there is even a remote chance that your application will need to be debugged...

Win++ provides a class for each type of display entity in the Windows environment. A base class exists for configurable objects and a number of derived classes are available for the most common types of Windows objects like dialog boxes. This provides a versatile method of generating objects. Instances of existing classes may be used to provide a quick method of implementing all the standard windowing features. A truly customised Windows object may be derived from the appropriate Windows base class.

Win++ does not provide an individual event handler for every possible event that could occur; instead there are a number of functions for supporting user-defined and other unsupported event types. There is also a function that handles the default action which can be used to provide a clean method of responding to an event before passing over control to the default procedure (eg trapping keyboard events).

Win++ costs $249 and the price includes source code. It is available from Blaise Computing Inc, 816 Bancroft Way, Berkeley, California CA94710. Blaise can be contacted on 0101 415 5405441.

Coming Soon

Microsoft is being unusually candid about its eagerly awaited C++ compiler, which will probably be shipped as C version 7.0 before the end of the year. The package will consist of a fully integrated PWB-like C and C++ programming environment for DOS and OS/2, with C++ support in the editor, a class browser and revamped CodeView debugger. The C++ implementation will be fully compliant with the AT&T 2.1 language specification, has been tested in-house for some time and is being used to develop the 32-bit graphics engine for Win 32 NT (due for release sometime next year).

Andrew King, Microsoft UK's Head of Languages, assures us that there will be extensive foundation and framework class libraries bundled with the product. These will be cosmic with all objects being derived from a superclass `C_Object`. Its 'AFX' GUI class hierarchy will only support Windows for the foreseeable future; the designers figure that if OS/2 2.0 runs Windows apps - why bother. If you are a faithful Microsoft disciple but just can't wait to get your hands dirty with C++, Microsoft suggests you get started with Glockenspiel's C++ and CommonView, as these class libraries are the nearest you'll get to what Bill's boys will be offering, and Glock code should be source-level compatible with the Microsoft compiler.

Borland's offering will be based on its Object Windows Library, which is presently only available for Turbo Pascal.

Conclusion

Including a GUI is not just a matter of adding the appropriate library and 'touching wood'. GUI-based applications tend to be notoriously power hungry; a static-linked library will add at least 100KB to the size of an application.

A 'turbocharged' workstation is normally required to run the application, but unsuspecting DOS users who may well expect the same application to run on their old IBM XT. Fortunately such end-users are few in number and have acquired the status of 'endangered species'.

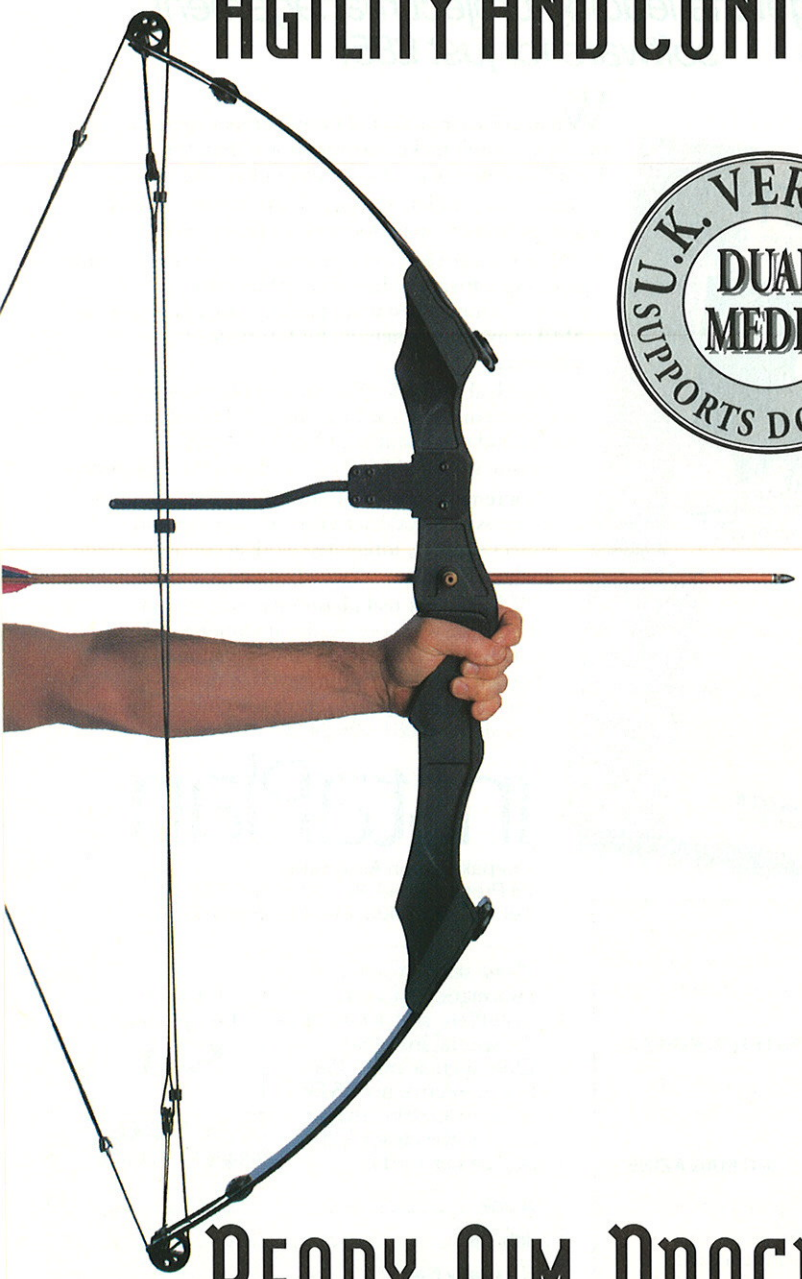
The GUI library is here to stay and the unwary software developer is already suffering from an embarrassment of choice. The rise in popularity of the GUI has started the ball rolling toward a new era in the design of more powerful, more friendly, end-user applications.

As a final word to you dinosaur software developers here's some sound advice, 'get on the bandwagon before it's too late!'

EXE

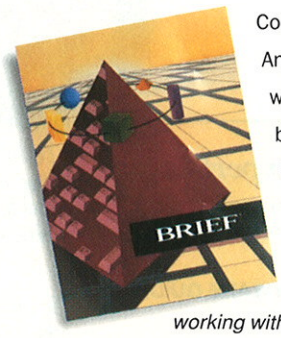
Paul Kemp and Cliff Saran are .EXE's latest luscious editorial acquisitions.

BRIEF'S NEW MOUSE SUPPORT GIVES YOU GREATER AGILITY AND CONTROL.



U.K. VERSION
DUAL MEDIA
SUPPORTS DOS & OS/2

More than 100,000 programmers around the world swear by the virtues of BRIEF – its unmatched power, flexibility, and intuitive command structure. Now, thanks to mouse support and other new features, version 3.1 of the best-selling text editor will make you quicker and more productive than ever. ¶ Hit a button on the mouse – a special, user-configurable menu window pops up and lets you execute any command you wish. Mark text and move around in your windows without using the keyboard. Zoom your windows to fill the screen. ¶ As snappy as it is, the mouse is only one enhancement among many. BRIEF 3.1 also takes advantage of EMS memory whenever it reads in a file. The result: faster performance, whether you're editing a file or running a macro. ¶ So what else is new? Redo – a step for step complement to undo. Support for Microsoft C 6.0 Advisor and Microsoft's Programmer's Workbench. And more. ¶ BRIEF's seamless interface with other Solution Systems products gives you instant access to the tools you need. A single keystroke lets you keep track of product development with Sourcerer's Apprentice – Version Control for the Professional. Analyze product performance with CHARGE. And nothing beats dBRIEF for tailoring BRIEF to Dbase or Paradox. ¶ *BRIEF sells for £199 excluding VAT.* ¶ *Once you start working with BRIEF 3.1 you'll really fly.*



READY, AIM, PROGRAM.

Order your copy of BRIEF today.

call:

0763 244141

Solutionsystems

THE PHYSICS OF PROGRAMMING

1 The Maltings, Green Drift,
Royston, Hertfordshire SG8 5DB
Telephone: 0763-244141
Facsimile: 0763-244025

CIRCLE NO. 063

NEW VERSION 3

Extra features include:

Plotter Support ■ WBS & OBS Coding ■ High-Resolution Charts ■ Proposal Format ■ Extended Printing Features ■ End Date Scheduling ■ Extended Importing & Exporting ■ Optional LAN Support

Save £460 when you upgrade to InstaPlan EMS

Now get the leading project management software for just £85!

YOURS FREE
Worth £14.95, this guide is yours whether you keep InstaPlan or not

Whatever your project planning or management needs, you can't make a better choice than the InstaPlan award-winning family of planning software.

Instaplan is widely recognised as offering the best combination of top performance with low price.

Now for just £85, you can upgrade from any existing planning software to InstaPlan EMS – the top-of-the-range InstaPlan package, supporting up to 8MB of expanded memory for full-scale project planning.

The deal is simple. You send us photocopies of your system disks, showing the serial number, and we'll send you InstaPlan EMS for just £85 – a fantastic saving of £460 on the normal selling price.

Supremely easy to use, InstaPlan EMS handles in excess of 10,000 activities and offers multi-project support, integrated work accounting, trend analysis and many other advanced features.

Why not get real planning power at your fingertips? With our 30-day unconditional money-back guarantee you've nothing to lose! For full details of this unbeatable money-saving offer, simply clip and return the coupon below, or call us on 081-908 2423.

InstaPlan

Deepak Sareen Associates,
28 Preston Road, Wembley HA9 8BR.
Tel: 081-908 2423. Fax: 081-904 6668.

Read what the experts say:

"Highly professional, highly competent, tremendous range of planning choice, terrific value-for-money."
GOOD SOFTWARE GUIDE

"InstaPlan is proof that simplicity of design and operation can be achieved without abandoning a good set of project management features."
PC MAGAZINE

". . . Useful . . . intuitive . . . practical . . . well worth the price."
PROJECT MANAGER TODAY

"InstaPlan offers the features of a £2,000 package with none of the trade-offs of its low-cost competitors."
PERSONAL COMPUTER MAGAZINE

"Technically it is extremely competent, being one of the only programs that is capable of levelling resources across multiple projects."
COMPUTER SHOPPER

"So simple to use that you wonder why other packages make a meal of resource allocation."
PC MAGAZINE

"Excellent rethink of requirements for project management packages . . . Ease of use – excellent, functionality and documentation – good."
PC USER

BEST PC MAGAZINE BUY

PERSONAL COMPUTER MAGAZINE EDITOR'S CHOICE

WHICH COMPUTER? BEST BUY

DON'T MISS OUT ON THIS SPECIAL UPGRADE OFFER

Complete and return the coupon today for full details on InstaPlan, plus this unbeatable upgrade offer.

Please send me a free information pack on InstaPlan, and details of the special InstaPlan EMS upgrade offer. Plus how to recrive my **FREE** guide to 'Getting the job done' (offer closes 30th September 1991).

Business Computing 91

FREE TICKET simply for replying

Name _____

Job Title _____

Company Name _____

Address _____

_____

Postcode _____ Tel _____

Do you currently use any project management software?

☐ NO ☐ YES If YES, please state which

_____

POST TODAY – NO STAMP NEEDED

DEEPAK SAREEN ASSOCIATES
FREEPOST, 28 Preston Road, Wembley HA9 8BR

Tel: 081-908 2423 Fax: 081-904 6668

.EXE 8/91

CIRCLE NO. 064

He called me 'Squire'

If a prospective customer sends Darrel Ince round to check up on your software house, you'd better polish your design documents, clean up your quality control and hide the aquarium.

One of the most enjoyable parts of my job involves visiting software companies. I am normally called in by a customer who wishes to employ a software developer, but has insufficient expertise to judge whether the developer they wish to use is competent. My job is to write a report and advise them. The process of evaluating a software developer can, at most, take a day although there are quite a few software developers who I have visited where it is easy to come to a conclusion after about ten minutes.

I enjoy this work because you come into contact with many of the Arthur Daleys of the software world; indeed, many of the people I meet make Arthur look like a paragon of morality and competence. I remember visiting one company where the director wore a snappy suit, had a massive Rolex watch (I suspect it came from Hong Kong) and smoked the size of cigar which you would associate with George Cole himself. My visit was a remarkable one for reasons which will be obvious later in this article, but also for the fact that it was the first time that I had been called 'squire' in my life.

The one slightly heartening thing for this country is that the US experiences are much worse

I asked him my standard set of questions there are about 25 which help me judge competence, and his company failed the first 23. His stock responses consisted of variations on 'What's that squire?' or 'Never heard of that one squire'. My twenty-fourth question is about training, and is normally accompanied by a request to see the training records. My aim in asking this question is to see what sort of courses, if any, the company sends its staff on. Whether they are sensible courses on design, project management and quality assurance, or courses on bit-twiddling in C, advanced bit-twiddling in C and structured bit-twiddling in C. When I asked my question he got up and made for the door as if to retrieve the training records. I was astounded: he had failed the first 23 questions abysmally, and here he was about to show me the training records.

He returned with something of a sad look on his face. 'Sorry squire', he said, 'bit of a disaster, we normally keep our training records on the computer and it's just gone down'. Presumably because he had just phoned from his secretary's office, telling the boys to pull the plug pronto. I commiserated, and asked to have a talk with whoever was responsible for training, a training manager perhaps. A lesser man would have admitted defeat. Not him. He stood up and left the office looking for his 'training supremo'. I spent five minutes wondering what the excuse would be this time. Perhaps he would say that his training manager had died that morning, possibly in a heroic (but vain) attempt to mend the computer.

He came back with a young man - perhaps half my age - who looked very flus-

tered. After the introductions I asked him whether he supported the National Computer Centre's HeadStart scheme. He assured me that his was one of the few companies in the area which fully subscribed to the ideals of HeadStart, was deeply committed to playing a major role in the scheme and he, this very day, was to ensure that line management were appraised of the virtues of the scheme. I found his commitment to the HeadStart scheme impressive and moving, particularly since it was something I had dreamed up on the spur of the moment. Clearly, the manager had left the office and found the first gull in his company to act out the fantasy of being training manager.

Something Fishy

A remarkable thing about this company, and many of the poor companies that I visit, was that it had a fish tank in the foyer. It seems almost axiomatic that the presence of a fish tank indicates awful software development practices, so that should be the first thing that you look for. The aim of the rest of this article is to point out some of the more technical signs of software malpractice.

One of the most important features of a competent software developer is the effort that is put into the system specification and the process of constructing this document. The system specification is the key document on the software project: the designer uses it to produce a system architecture; quality assurance staff use it to derive the system and acceptance tests; the project manager uses it normally in outline form, to produce a costing for a project; and the technical authors use it to produce user documentation. Get the specification wrong and everything is wrong. There are a number of natural laws in this world, for example, the sun comes up in the morning and sets in the evening. There is also the natural law that if a sentence in a document is capable of being interpreted in n ways and n people read that document, then each will interpret the sentence differently. This means that a poor specification will give rise to a system which does not implement what the customer requires; which is validated by tests that bear little relationship to live data; that is produced by a project which goes over budget; and which is accompanied by user documentation that bears more resemblance to a work of fiction than a user-oriented view of the properties of the system.

A correct concern with the system specification manifests itself in a number of ways. The major indicator is that the software developer spends at least 20% of his resources in analysing and writing down user requirements. Another indicator is the degree to which validation occurs during the early stages of the project when the system specification is produced. If the developer uses a fourth generation language then prototyping should occur, and there should be technical meetings, such as reviews, to which the customer has been invited. The developer should also have a quality standard which insists that his staff produce the system tests as soon as possible after the system specification has been completed. There's nothing quite like the discipline engendered by a good tester looking over your shoulders at the system specification saying 'But how the hell do I test that?'

Someone who hates the Project Manager will come up to me later and say 'Psst, they are in the cupboard'

Beware, however, of the software developer who claims to use prototyping. One of the first questions I ask when visiting a dubious software developer is about the standards used for producing system specifications. A common answer to this question is 'We don't have a system specification guv, what we do is to produce the system and show it to the customer and then change it, and so on, until everybody is happy - it's a modern technique known as prototyping'. Beware of this attitude: it's just old-fashioned hacking masquerading beneath the fashionable garments of a modern technique. Even companies that carry out prototyping still produce a system specification after the prototype has been agreed with a customer; for one thing it is vital for maintenance.

A technique which I've found useful when asking questions of a rogue software developer is to wear a bow tie. It's also useful if you have some form of academic title such as 'Dr' or, even better, 'Professor'. It also helps to develop a distracted, rather abstract, academic air. In the early years of my marriage I developed such an air in order to avoid tasks which my wife assigned me, such as cleaning out the garage: 'Garage, garage, now where would I find that'.

The first thing that I do when visiting a suspect software company in a large city such as London or Manchester, is to admit to having major problems finding the city. Not the street in which the company is situated, or even the district, but the city. The look of relief that crosses the face of the staff that I meet, when confronted by a bow-tied academic who is incapable of finding a major conurbation is totally visible. They become relaxed, they make no effort to hide the various warts and carbuncles in their development environment, and assume that the absence of any standards, procedures and management practices will pass by a visitor who is clearly incapable of finding his way out of the building. It's only when you start quoting the page numbers of BS5750 and talking about NATO defence standards do they realise that instead of swimming with a rather amiable, distracted porpoise they have found themselves with a basking shark, but by then it's too late.

Independence

One set of questions that I ask a software developer concerns independent validation. That is, the extent to which the checking of a software product, a specification, a design or a software component is carried out by staff who have not produced the product, or who do not have a vested interest in the product passing a validation check. For example, many programmers do not like discovering errors in their code because it reminds them of their mortality. Consequently, the testing of subroutines or partial versions of a system is often skimped. Also, managers are so obsessed with budgets and handover dates that activities such as system testing are badly skimped. If a software developer uses independent validation teams, or insists that modules are tested by programmers who have not produced the module, then that is an excellent sign.

A bad sign is a flag flying outside the developer's building announcing the fact that the company has been given the Queen's Award to Industry. In my experience, there is no correlation between this and software competence. Indeed the reverse is often true. Certainly, if I see this flag flying outside a developer's building *and* a fish tank inside, then I know that it is not even worth my while asking any technical questions of the software developer.

RECESSION SPECIAL!

PRE-SALE CONSULTING

TAKE 15% OFF ALL PRICES!

POST SALE SUPPORT

NEW & NOTEWORTHY

Drover ToolBox for Windows v1.2 .....£214
A dynamic link library of 200 functions and seven classes for Windows 3.0 development.
Lahey F77L v5.0 .....£511
New features include allocatable arrays and other important Fortran 90 features.
LANtastic 2Mbps Starter Kit £319
Economic alternative to Novell. Links as few as two PCs and grows upto 100s.
MS CD-ROM Programmer's Library £243
Contains over 70 indexed and cross referenced books and reference manuals, with over 20MB of code that can be included in your programs.
MS Excel v3.0 for Windows £199
Incredible price! Order while supplies last!
MS MS-DOS v5.0 Upgrade £69
Loads into extended memory. Network drivers, TSRs in upper memory. Full screen editor, Undelete, Unformat and more!
MS Visual BASIC £129
Create GUIs in minutes without writing a line of code! Then add functionality with a powerful general-purpose language!
NE-2000 Ethernet Kit £154
16-bit high performance network adapter.
Novell NetWare v2.2 .....£ 556
Combines Netware 2.x, SFT, Advanced and ELS I & ELS II into one product.
Oracle Tools & Database v6.0 .....£674
Now includes PL/SQL and SQL*DBA.
Viva 2496if 9600bps FAX/Modem £159
9600bps int. Send/receive fax/modem.
Zortech C++ Developer's Edition v3.0 ...£350
Optimizing C++ Compilers for DOS, Windows & OS/2

ARTIFICIAL INTELLIGENCE

Arity Combination for Windows £834
PDC Prolog (Turbo Prolog) £201

ASSEMBLERS & LINKERS

Blinker £201
Dis/Doc Professional £188
Microsoft Macro Assembler £97
Sourcer £105
Spontaneous Assembly £160

BASIC

Mach 2 for MS QuickBASIC £55
Object Script QuickBASIC £105
PowerBASIC £71

BUSINESS

ABC Flowcharter £177
Adobe Illustrator for Windows £271
Baler £308
Form Tool Gold £55
Grammatik IV £54
IBM Current £213
IBM Storyboard Live £276
Micrografx Charisma £324
ORG PLUS Advanced £60
Quattro Pro Competitive Upgrade £88
Show Partner F/X £224
SideKick Plus £125
Super Project Plus £238
Xerox FormBase /Windows £297

C

Aztec C Cross Development Systems £376
Borland C++ v2.0 .....£295
C Communications Toolkit £129
C-terp 386 with Source £192
CLEAR+ for C £120
Control Pak/Windows £383
DESQview API C Library £110
Essential Communications £194
Greenleaf Comm++ £149
Greenleaf ViewComm £420
Informix C-ISAM for DOS £147
Lattice C Development Sys v6.0 .....£143
Object Script C £349
PC-Lint £75
Pro-C with Workbench £557
Turbo C TOOLS £96
Turbo C++ Second Edition £63
WATCOM C /386 Professional £904
WindowsMaker £596
XVT (Windows/ PM/ Macintosh) £415
Zortech C++ Database Class Library £193
Zortech C++ for Windows & DOS £196

CADD

3D Drafting £199
AutoSketch £143
Design Cad 2D £145
Design CAD 3D £193
DRAFIX CAD for Windows £410
DRAFIX CAD ULTRA £237
Generic CADD £281

CASE

Case: PM for C++ £1,713
Case: Windows Corporate Edition £855
Case: Windows Standard Edition £425
Competitor II £225
EasyCASE Plus v3.0 .....£425
Oracle CASE Dictionary £1,504

COBOL

COBOL spII £415
COBOL/2 Compiler with Toolset £1,449
Dialog System £805
Level II COBOL £398
RM/COBOL-85 Development System £756

COMMUNICATIONS

Carbon Copy v6.0 .....£108
Crosstalk for Windows 3.0 .....£117
LapLink Pro v4.0 .....£84
Mirror III £81
Procomm Plus £64
Reflection 1 for Windows £289
Remote2 (Host & Caller) £105
SmarTerm 240 .....£202
WinConnect £58

DATABASE & FILE MANAGEMENT

Advanced Revelation £705
ALPHA/Four competitive upgrade £91
AskSam £213
Btrieve DOS 3.1 Network for PC £383
Btrieve for Windows £383
Clarion Personal Developer £43
Clarion Professional Developer £444
DataFlex £393
dbFast/PLUS £166
dbFast/Windows £239
FairCom Toolbox Special Edition (1-user) .£448
Gupta SQLWindows £910
Informix SQL for DOS £521
Oracle Tools & Database v6.0 .....£670
Paradox v.3.5 .....£488
Paradox SQL Link £335
R&R Clipper and FoxBASE+ Module £42
R&R Relational Report Writer for dBASE .£160
R&R Report Writer for Oracle SQL £192
R&R Worksheet Report Writer £96
R:BASE Single User £469
SQLWindows for Btrieve £1,370
Superbase 4 Windows £418
UI Programmer £320
XQL Relational Data Manager £529
Xtrieve Plus £403

DEBUGGERS

ClearView-386/486 Debugger £318
Multiscope Debugger for DOS £114
Multiscope Debugger for Windows £240
Periscope II with Breakout Switch v5.0 .....£121
Turbo Debugger & Tools v2.0 .....£97

DESKTOP PUBLISHING

Adobe PostScript Cartridge £297
Adobe Type Align /Windows £59
Adobe Type Manager for Windows £60
CatchWord OCR £136
ColoRIX VGA Paint £97
Facelift for Windows £58
Fontware Installation Program £56
Fontware Typeface Packages £77
Go Script Plus £179
Inset + Hijaak £160
OmniPage Pro £598
OmniProof for DOS £99
PageMaker £493
Persuasion for Windows £303
The Complete Page Scanner w/ OCR £579
WordScan 286/386 2 Mb £383
Xerox Ventura Publisher Gold v3.0 .....£504

DISK/DOS/BACKUP

Back-It £74
Copy II PC £26
Disk Manager £35
Norton Commander £94
Norton Utilities £110
OPTune £43
PC Tools with AntiVirus £97
Q-DOS II £43
SpeedStor Universal £61
SpinRite II £58
WonderPlus £47
XTree Pro £54
XTree Pro Gold £89

EDITORS

Brief v3.1 .....£176
EDT + ...£213
KEDIT £97
Norton Editor £47
Sage Professional £189
VEDIT PLUS £110

FORTRAN

Fortran Toolkit 2 .....£116
GRAFMATIC & PLOTMATIC £155
RM/FORTRAN Development System £345
Spindrift Utility Library (F77L) £79
ULI Maths Functions Library (F77L) £188

GRAPHICS

Corel Draw £374
Designer for Windows £455
Dodot £111
Draw Applause £263
GRASP with ARTOOLS £191
GSS GKS Kernal System for DOS £415
GSS Graphic Devlpnt Toolkit for DOS £555
HALO £227
HALO Professional £341
Halo Windows Toolkit £338
Harvard Graphics £301
HiJaak £128
MetaWINDOW £222
PCPaintBrush for Windows £52
Pizazz Plus £85
Windows Draw Plus £160

HARDWARE

8087-10 Math Coprocessor £157
ASCII Internal 2400 Modem £68
ASCII Internal 9600bps V.32 with MNP 5 ..£428
FastMath 83S87-16 Math coprocesor £243
Hardcard IIXL 105Mb, 9ms Hard Disk £555
Logitech TrackMan Serial Mouse £76
MouseMan Serial Mouse £68
Pacific Page PS Cartridge for HP IIP £348
Paradise 1024x768 VGA w/ 512k £191
Practical 2400 Pocket Modem £106
ProScan PS1000 64 Grayscale Scanner £520
SatisFAXtion Classic £375
ScanMan Plus PC £171
The Complete Communicator £412
The Complete Page Scanner £530
VIVA 9600 V42 bis External Modem £446
Weitek 3167/25MHz Math Coprocessor £558
Worldport 2496 Portable Fax £465

INTEGRATED

Framework III £424
Lotus Symphony £485
Lotus Works £89
Microsoft PC Works £92

LAN

LANtastic AE-2 Ethernet Starter Kit £450
NE-1000 Ethernet Kit £125
Netware v. 3.11 20 user £2,327
Netware C Interface for DOS £196
Oracle for Server £1,254
Procomm Plus for Network £325
Q-DOS II for Network (8-user) £134
Saber File Manager £200
Saber LAN Administration Pack £351
Saber Menu for MS Windows 3.0 .....£249
Saber Menu System Single-User £45
Saber Meter Usage Tracking Software £99
Saber Secure £75
SoftwareBridge LAN £143
Super Project Expert Lanpak £420
Superbase 4 Windows Network £598

MAC

Adobe Illustrator Mac £332
Adobe Type Manager £56
Carbon Copy Mac £108
LapLink Mac III £84
MathCad £305
Mathematica £425
Norton Utilities for Mac £82
Oracle Developers Kit Mac £501
Pagemaker £486
PC-Matlab /Mac £752
Persuasion £311
Quark Xpress £512
Smalltalk/V for Mac £107
Zortech C++ for MAC £196

MEMORY MANAGEMENT

386 Max £60
Above Disk £58
Headroom £70
QEMM 386 .....£57
Turbo EMS £48
Vcache £45

MODULA-2

EmsStorage with Full Source £48
Logitech Modula-2 Development System ..£158
Logitech Modula-2 Window Pack £30
ModBase £21

OBJECT ORIENTED PROGRAMMING

C++ Views £319
Codebase ++ £222
HCR/C++ £727
Smalltalk/V £69
Smalltalk/V 286 .....£107
Smalltalk/V Communications £27
Smalltalk/V EGA/VGA Colour Pack £27
Smalltalk/V Goodies 1 .....£27
Smalltalk/V-Windows £268
Zinc Interface Library £160
Zortech C++ MS-DOS 386 .....£641
Zortech Windows Toolkit £45

OPERATING ENVIRONMENTS

386/ix Operating System Release 3.2 .....£330
DESQview £77
DR DOS £130
DR Multiuser DOS £448
GeoWorks Ensemble £105
Microsoft Windows £92
OS/2 Extended Operating System £662
OS/2 Standard Operating System £258
SCO UNIX System V/386 v.3.2.2 .....£370

OS/2

Corel Draw for OS/2 .....£320
Describe Word Publisher (Client) £164
Display Write 5/2 .....£234
Halo for OS/2 .....£338
Microsoft OS/2 Presentation Mgr.Softset £92
Microsoft OS/2 Presentation Mgr.Toolkit £308
Oracle OS/2 .....£652
PageMaker for OS/2 .....£486
Paradox OS/2 .....£462
Qedit for OS/2 .....£48
QEMM Expanded Memory Manager 50/60 .£58
Smalltalk/V-PM £268
Super Project Expert/2 .....£481
ToolBook (OS/2) £280
TopSpeed C++ for OS/2 Standard £213
Vitamin C for OS/2 .....£319
Wide Angle for OS/2 .....£62

OTHER

ACT! (Sales lead tracking) £233
Adlib Card (Sound) £104
Builder (Batch Compiler) £81
graphics-MENU (GUI) £134
Guide 3 for PC (HyperText) £358
HP NewWave (GUI) £105
HyperPAD £82
Install (Automatic installation) £148
KnowledgePro £420
Menuet (GUI Development) £209
PolyAWK v1.3 .....£126
RPG II Development System £1,117
SCHEMA-PCB (SC-4) £733
SoundBlaster £147
System Sleuth (Diagnostics) £72
VirusCure Plus £64

PASCAL

Asynch Plus £121
B-Tree Filer £97
Graf/Drive Plus Personal £81
Object Professional £132
Pascal TOOLS £81
POWER TOOLS PLUS £96
Topaz £64
Turbo Analyst £90
Turbo Pascal for Windows £150
Turbo Pascal Professional £190
Turbo Professional £90

SCIENCE & ENGINEERING

ChiWriter Chemistry Support £45
Derive v2.0 .....£183
InGraf £263
Labtech Notebook £801
Matfor - 286 .....£225
MathCad for Windows £303
Mathematica 386 .....£597
Mathematica for Windows £855
PlotLib £188
SSP/PC Scientific Subroutine Library £245
Tech*Graph*Pad £335
Zortech C++ Science & Engineering £490

SCREEN DESIGN & MANAGEMENT

C Windows Toolkit £86
Flash-Up £69
Hi Screen XL £112
POWER SCREEN £96
ProPak £412
SayWhat!? £33
Vermont Views £319
Vitamin C £254
WNDX GUI Development System £321

TRANSLATORS

BAS_C Commercial £242
FOR_C ++ £870
FOR_C with Runtime Source £628
Pascal to C Translator £140
Turbo Plus £138

UNIX/XENIX

Architech Series Workstation Developer..£1,528
DataFlex for UNIX 386 .....£950
Informix C-ISAM (SCO 4.0) £295
LPI COBOL Compiler £1,076
Norton Utilities for System V v3.2 .....£174
Oracle UNIX/Xenix 386 .....£1,476
SCO UNIX System V/386 Dev.System £641
VEDIT PLUS for UNIX/XENIX £184

WINDOWS DEVELOPMENT

Asymetrix ToolBook £280
BeckerTools £73
Microsoft C with Windows SDK £466
MS Windows Device Driver Dvlpnt Kit £311
Microsoft Windows Productivity Pack £38
Object Vision £86
WIN++ £160
WindowsMaker Professional £746
WinSleuth £72

USA ★ SOFTWARE

2 OLD BROMPTON ROAD • SUITE 315 • LONDON SW7 3DQ
PHONE: 071–351 2360 • FAX: 071–352 0974

Over 3,500 products carried. Call today for price list-disk.

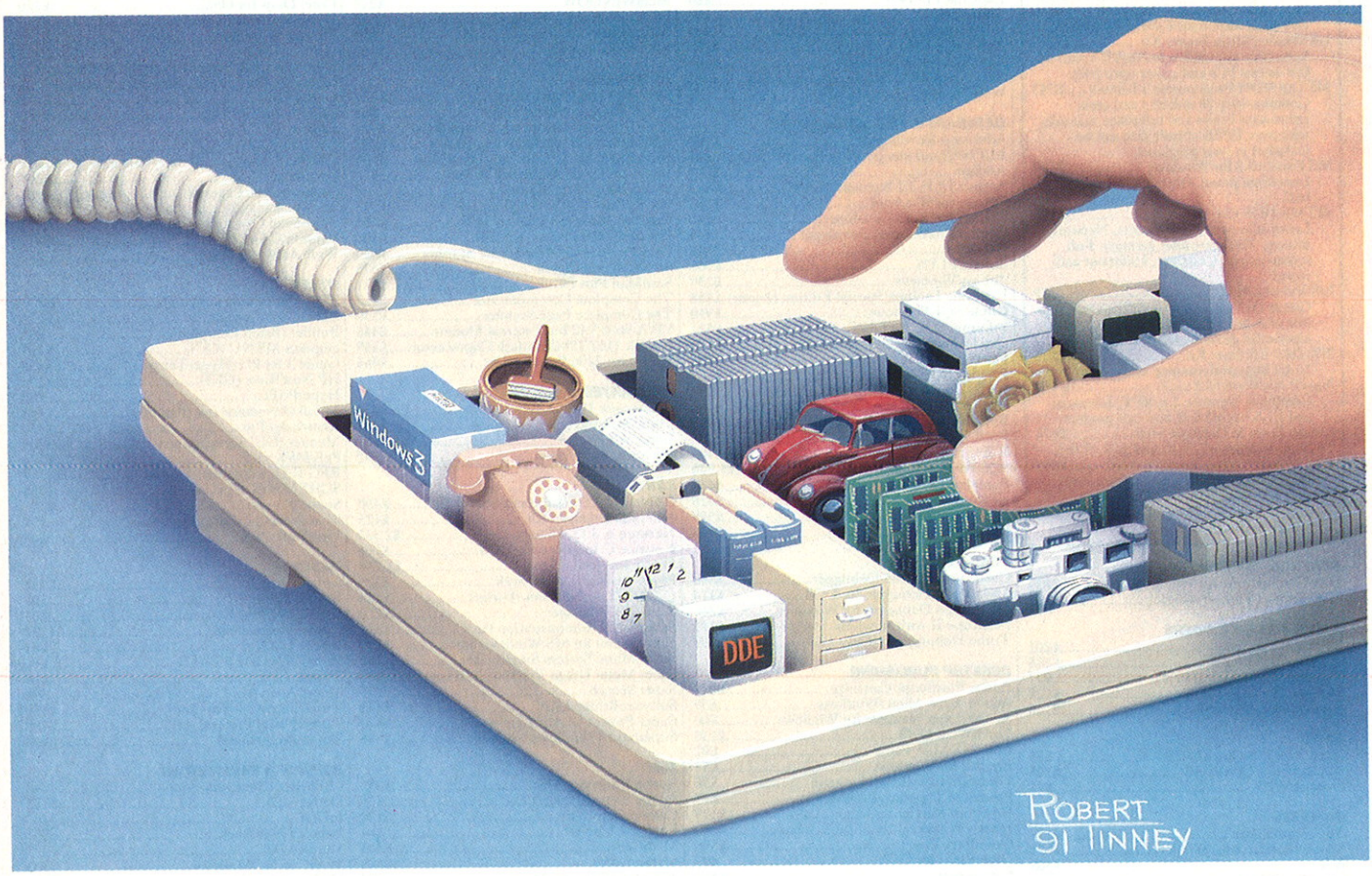
We will match any advertised Price!

- Prices exclusive of V.A.T.
- All major credit cards accepted.
- 30 day invoicing for qualified organizations.
- Special discounts for large or corporate buyers.
- Prices subject to change without notice.

UK8/91

CIRCLE NO. 065

Objects At Your Fingertips.



Now, if you want to develop applications for Windows 3.0, there's a fast and easier way to do it with the premier object-oriented programming language. Smalltalk/V.®

With Smalltalk/V Windows, you can explore, prototype, build finished applications and ship them runtime free.

You can tap into applications using DDE so effortlessly you don't have to be a Windows expert to do it.

And with one of the world's most comprehensive class libraries, you can choose our objects or easily build your own.

But whatever you develop, it will be portable between the Windows, OS/2 and Mac versions of Smalltalk/V.

With so much at their fingertips, more people are solving more problems with Smalltalk/V than any other object-oriented programming system.

At only £330 + VAT and no runtime charges, you can solve them, too.

Just call us at 071-436 9481. And see why programming Windows has never been easier.

Smalltalk/V Windows

COCKING & DRURY (SOFTWARE) LTD

180 Tottenham Court Road, London W1P 9LE Phone 071-436 9481 FAX 071-436 0524

CIRCLE NO. 066

Life-cycle mantra

Another set of questions that I ask a software developer concerns change. Often I visit software projects and am shown around by a senior manager. He usually takes me to see some of the more prestigious projects that they are undertaking. When visiting a project which is in its last stages, say system testing, I usually ask the manager who on his staff is still carrying out requirements analysis. The answer is usually in the form of a mantra or chant: 'We are a company totally devoted to software quality assurance, we have a phased life-cycle model where our projects are split up into a series of distinct phases. Each phase delivers a document which then forms the input to the next phase, this subsequent phase then produces another document which again is passed on to the next phase and consequently no staff are involved in this activity'.

My reaction is to admire the ordered way in which the software projects are organised and then ask my question again. The answer I receive is: 'We are a company totally devoted to software quality assurance, we have a phased life-cycle model where our projects are split up into a series of distinct phases...'. However, I usually take the precaution of asking the question loudly, since there will often be a member of staff, usually someone who hates the project manager, who will come up to me later and say 'Psst, they are in the cupboard'.

They are in the cupboard for a number of reasons. The first reason is that software development is probably the most difficult task that we carry out. Consequently, errors are going to be made which only surface late in a project, and which require re-specification and redesign at quite a late stage in that project. Another reason is that for any project which delivers more than a trivial system, changes in requirements are going to hit that project almost from the first day on which the project has been set up: the project which delivers an accounting package will be affected by changes in tax laws, the project which produces a management information system will be affected by a new management team taking over in the customer's company and the defence system project will be affected by changes in the technological capability of a potential aggressor, whoever that is these days.

Because of the dynamic nature of software projects, the competent developer will have set up a formal system known as a configuration management system which evaluates whether a change is necessary, sanctions the change, checks that the change has been correctly applied and then lets everybody know what has happened. This will avoid the major problems that, for example, be-devils American defence projects.

Presumably he had just phoned from his secretary's office, telling the boys to pull the plug pronto

Traceability

Another question I ask a software developer concerns something known as 'traceability'. I normally ask the developer to show me the program code for a system which has been completed. After admiring the programming standards I place my finger on a section of code, usually a subroutine or procedure if the language is a third generation one, and ask them which functions in their system specification that code helps implement. If they come back quickly, say in five minutes, and tell me, that for example, the code helps to support the updating of a sales order file and also supports the updating of a warehouse file, then the developer is very well organised. If it takes around an hour the developer is usually competent. Anything longer usually indicates incompetence. The response 'What's a system specification?' means that you have hit rock bottom.

Traceability is vital for a number of reasons. Let us examine one: that of rerunning acceptance tests. Acceptance testing is the process whereby the developer and the customer run a series of tests on a system which demonstrates that it actually meets the requirements of the user. Acceptance testing is one of the most nerve-racking activities on the software project, if anything goes wrong during this activity it can take a lot of resources to put it right. However, it is not unknown in the software industry for an acceptance test to fail. A responsible software developer will then try to discover what exactly has gone wrong, and modify the system to get rid of the problem. After rectification, the acceptance test is rerun to check that the modification to the system has been made. A poor developer will then carry on testing, so for example, if acceptance test 1228 has failed and been rectified, the developer will then execute acceptance test 1229. Wrong. If there is no traceability in the system documentation the testing has to start again from the beginning.

The reason for this is that in modifying the system in order for it to pass an acceptance test the programmer may have modified a module which is also executed by a previous acceptance test. Now, industrial studies have shown that making a modification to a system at a late stage in a software project is a risky business. The figures seem to suggest that a change applied late in a project only has a chance of being correct with a probability ranking from .2 to .5. The probability will depend on factors such as the competence of the programmer, the programming language used and the degree to which debugging tools were used. Even with all the factors favourable it seems that the probability of an error occurring will still be as high as .5.

This implies that a change to a system to make it pass a particular acceptance test will often mean that it could fail the series of earlier acceptance tests. In order to guard against this, a developer should keep track of which modules are executed when a particular acceptance test is exercised. This means that only those previous acceptance tests which involve the modified modules are rerun. If this traceability documentation is not kept, then all the acceptance tests have to be rerun. Indeed, I always advise clients who sign a software contract to put the phrase 'the customer reserves the right to ask for the full acceptance test suite to be run after a rectification following an acceptance test failure' in that document, it usually scares the hell out of the software developer.

This is just one reason for having traceability in the documentation used by a software developer. The software engineering text books describe many other reasons which, when added up, means that if traceability is not built into system documentation then a developer will experience major problems.

25 Questions

These, then, are some of the questions which I ask a software developer. As I

have said, I have a list of 25 and, on average, most competent software developers can answer 15 of them well. Unfortunately, something like 40% of the developers that I meet have a hard time in answering two or three correctly. The computing industry is relying more and more on consultants or independent validation organisations who use a standard set of questions to dig out any incompetence among software developers. The major player in this area at present is the American Department of Defense, which uses a questionnaire of about 65 questions to root out companies who should not be contracted by them. This questionnaire was developed after the American government discovered that out of all the software projects contracted by them only 2% of the software was used as delivered, with 47% delivered but never used and 30% paid for but not delivered.

The one slightly heartening thing for this country is that the US experiences are much worse than mine. Eighty-five percent of the companies they have surveyed in the defence industry have not reached the barely adequate stage with only 1% of the companies wholly satisfying the surveyors.

It seems almost axiomatic that the presence of a fish tank indicates awful software development practices

The major experts on software capability assessment in the United Kingdom are housed in the Institute of Software Engineering, a government-funded body based in Northern Ireland. They carry out a free assessment of your software capability based on a 60 point questionnaire which you have to fill in. They provide you with this assessment in terms of a points scoring system which tells you where you lie in a spectrum that ranges from 'appalling' to 'very competent'. Give them a ring or send them a letter, as part of their brief is to improve the capabilities of British software companies. But get rid of the fish tank first.

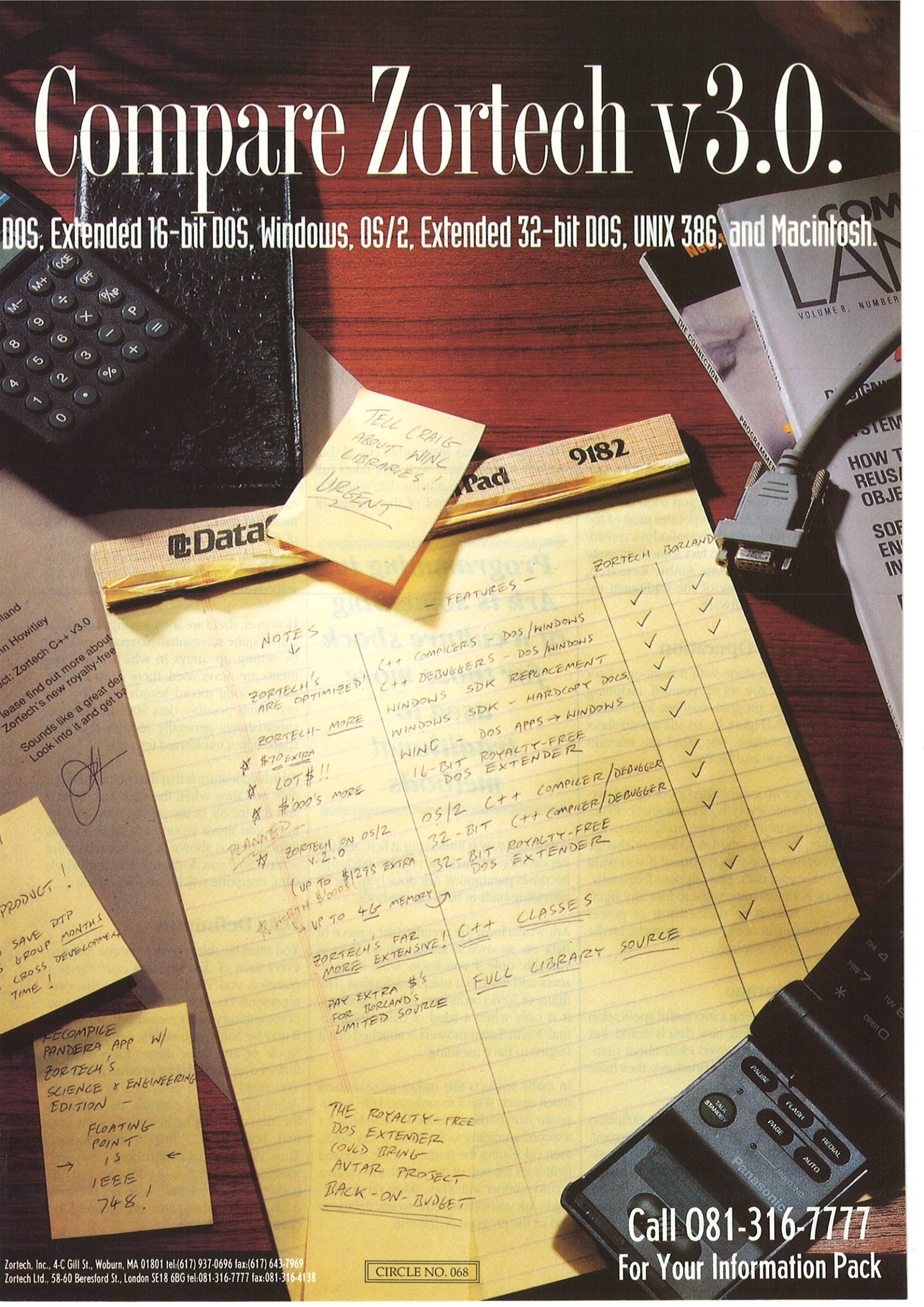
EXE

Darrel Ince is a Professor of Computing Science at the Open University. He will be sending a large sum of money, in used unmarked tenners, to the Editor of .EXE, in return for which we will not reveal his personal telephone number to the Aquarium Society of Great Britain [Telephonic Abuse division].

The Institute of Software Engineering can be found at 30, Island St, Belfast BT4 1DH. The manager in charge of software capability assessment is Ken Thompson - 0232 738507.

Compare Zortech v3.0.

DOS, Extended 16-bit DOS, Windows, OS/2, Extended 32-bit DOS, UNIX 386, and Macintosh.



Call 081-316-7777 For Your Information Pack

Zortech, Inc., 4-C Gill St., Woburn, MA 01801 tel:(617) 937-0696 fax:(617) 643-7969
Zortech Ltd., 58-60 Beresford St., London SE18 6BG tel:081-316-7777 fax:081-316-4138

CIRCLE NO. 068

Preserving Data with Ark

What's this? A spreadsheet-like thing in .EXE? Have the suits finally won? Indeed not - experienced programmers can develop substantial applications with the unusual Ark program, as Stephen Morris explains.

At first glance, Ark looks like an ordinary spreadsheet: titles around the border, columns of data, a strip of pull-down menus. Delve a bit deeper and you soon discover that, although Ark incorporates most of the features you would expect to find in a spreadsheet program, it is in fact something quite different. Ark, from Ambit Research, doesn't fit into any of the traditional categories of PC software.

Program Operation

The program starts with a front-end User Menu. This you set up yourself, including worksheets, program routines and further sub-menus (Figure 1). To edit a worksheet or run a routine, just select the relevant menu option and hit Enter.

There are no complicated loading and saving options - everything is automatic. Every time you move from one worksheet to another, escape to the User Menu or change program mode you are asked if you want to save your data. This happens frequently, so it is almost impossible to lose any significant amount of data - short of power failures, lightening strikes, magnetic media errors, crass stupidity...

Data Storage

The key to creating a successful application is in understanding how data is stored. Put aside any preconceived ideas about traditional database record structures; they have no place in this package.

As far as Ark is concerned it is the definition of the data, not its position, that is important. Each application - which can incorporate many dozens of worksheets - holds all its data in a single 'pot'. This is a DOS file, with no recognisable structure but with an identifying tag attached to each item of data.

This approach may seem rather chaotic but actually it works very well. Ark takes full responsibility for finding the data when it is needed. Losing control of the storage of data in this way is unnerving at first, and the importance of making regular back-ups becomes paramount! Ark does help out by providing built-in back-up methods.

Programming in Ark is something of a culture shock for those more used to traditional methods

Ark works from the premise that a piece of data on its own is meaningless. Data is valuable only if you know what it represents and if it has a unique definition. The figure of 70553 on its own tells you nothing. It is only when a label of Sales/1989/Actual/North/Lawn-mowers is attached that it begins to have meaning.

In many respects this makes programming much easier. With a record-based file, a particular item of data is found by searching for a record according to the value of key fields, then calculating the position of the required field from the start of the record; this yields a record number and offset. When programming in Ark you stick with the wordy definition and let the program do the rest.

Ark maintains a dictionary in which are stored the descriptive names that you use. When the data is stored, the tag exists physically as a sequence of codes that identify names in the dictionary. Of course, the lack of file structure does create a considerable overhead - each data value needs a minimum of 30 bytes. An application of reasonable size will consume notable amounts of disk space.

However, there are some savings and these can be quite substantial. No space is wasted by setting up arrays in which many elements are never used; there is no need to allocate long record lengths to allow for maximum possible data size. In practice, spreadsheets generally require less disk space when transferred into Ark.

An added feature is that the tag includes the date and time when the data was created and the initials of the person who entered it. As you move around a worksheet, this information is shown at the bottom of the screen. So if you do change someone else's data, everyone will know who dunnit.

Data Definition

You must be able to give a unique name to every item of data. In each case, the first question you ask is: What does this data represent? The answer gives you the name of something you are measuring - the entity. It may be 'Sales', 'Quantity', 'Age', 'Opening Stock' or any other name. Each entity is further described by one or more attributes. You need enough attributes to uniquely define the data. For example, the attributes for Sales data may be Period, Analysis (Actual, Forecast etc), Region and Product. Thus '1989', 'Actual', 'North' and 'Lawn-mowers' are all examples of attributes and this particular group identifies one item of Sales data. Other Sales items are defined by changing the attributes.

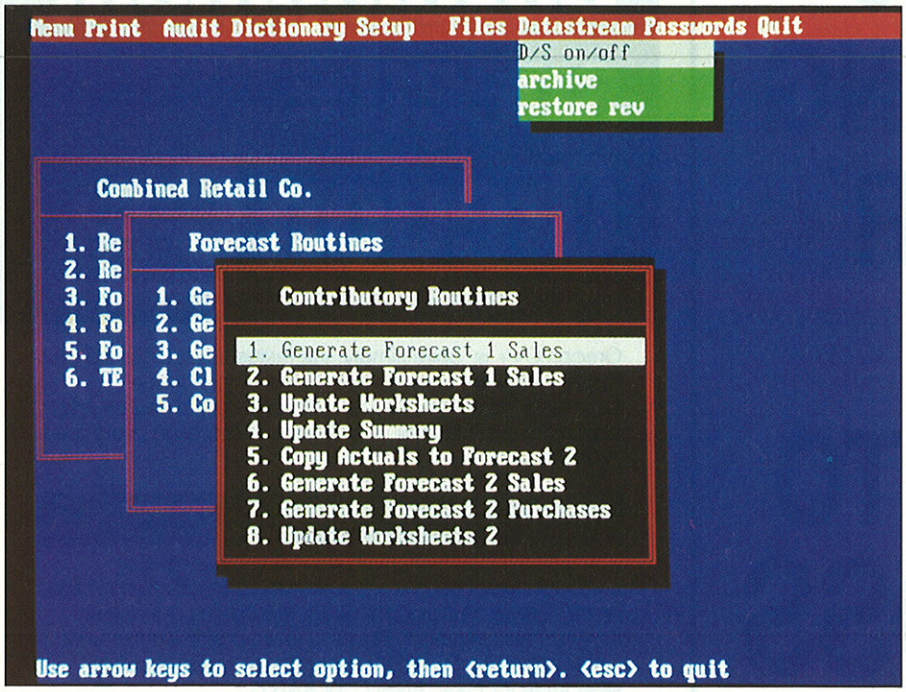


Figure 1 - An Ark User Menu

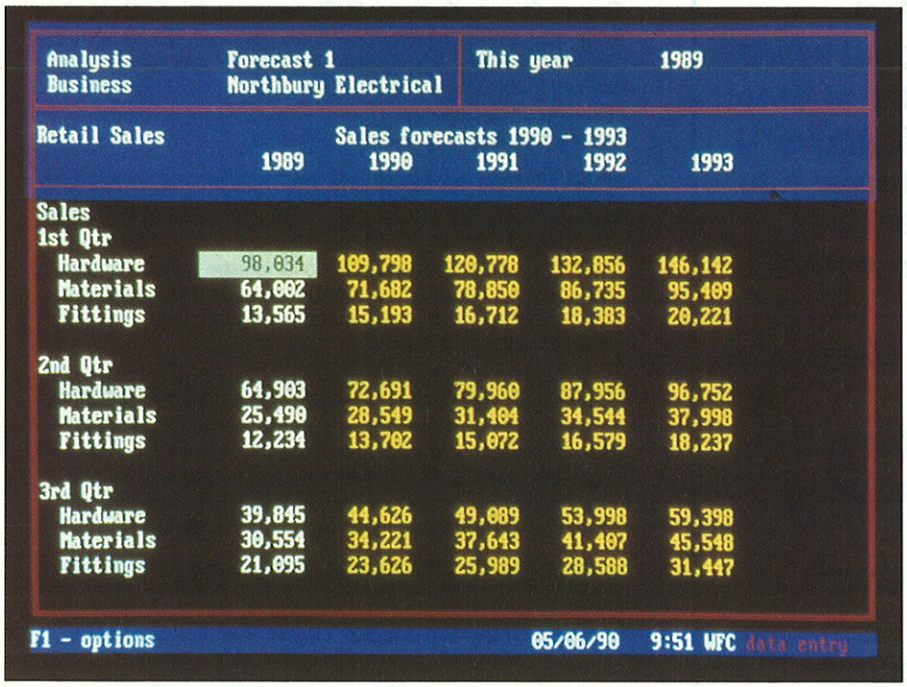


Figure 2 - An Ark worksheet

Perhaps the best way to start is by thinking of the data as being stored in multidimensional arrays. Each entity name is analogous to an array name. The Sales entity name is equivalent to a four-dimensional array. If the application covers five years, three analyses, four companies and 80 products, a conventional array structure would require you to set aside enough space in memory for 4,800 values - and a similar amount on file. With Ark, you need set nothing aside; data is read from memory as and when it is needed.

Surprisingly, perhaps, the hard part is in deciding just what is an entity and what are its attributes. Once you start to analyse the data you will find that several possible structures spring to mind. It is not a disaster if you don't get it right first time - Ark has facilities for adding new attributes or taking away superfluous ones - but often you have no alternative but to throw it all away and start again. However, the second time around you can build your application in a fraction of the time.

This definition stage is all about understanding your data and knowing how it relates to the real world (a tricky concept for programmers at the best of times!). It can also require some interesting lateral thinking. It may seem that a Revenue item from a shop should be defined by the Sales Assistant, Shop and Town (amongst others) but, if each Sales Assistant is uniquely coded or named, the other two attributes are not necessary. A Shop can be defined as a collection of Sales Assistants and a Town as a group of Shops.

Creating Worksheets

Once the data definitions have been sorted out you can get down to the business of setting up the worksheets, on which data will be entered, interim calculations performed and reports generated.

Again, the approach is the reverse of more traditional methods. On a normal spreadsheet the data is positioned first and labels are added above and to the left, to serve as a reminder of the function of the data. For example you may have column headings for Opening Stock and Sales, with the 80 product names listed down the left-hand side. This will give you a sheet for a particular period, analysis and company. To cater for all five periods, three analyses and five companies you would need 60 copies of the spreadsheet file.

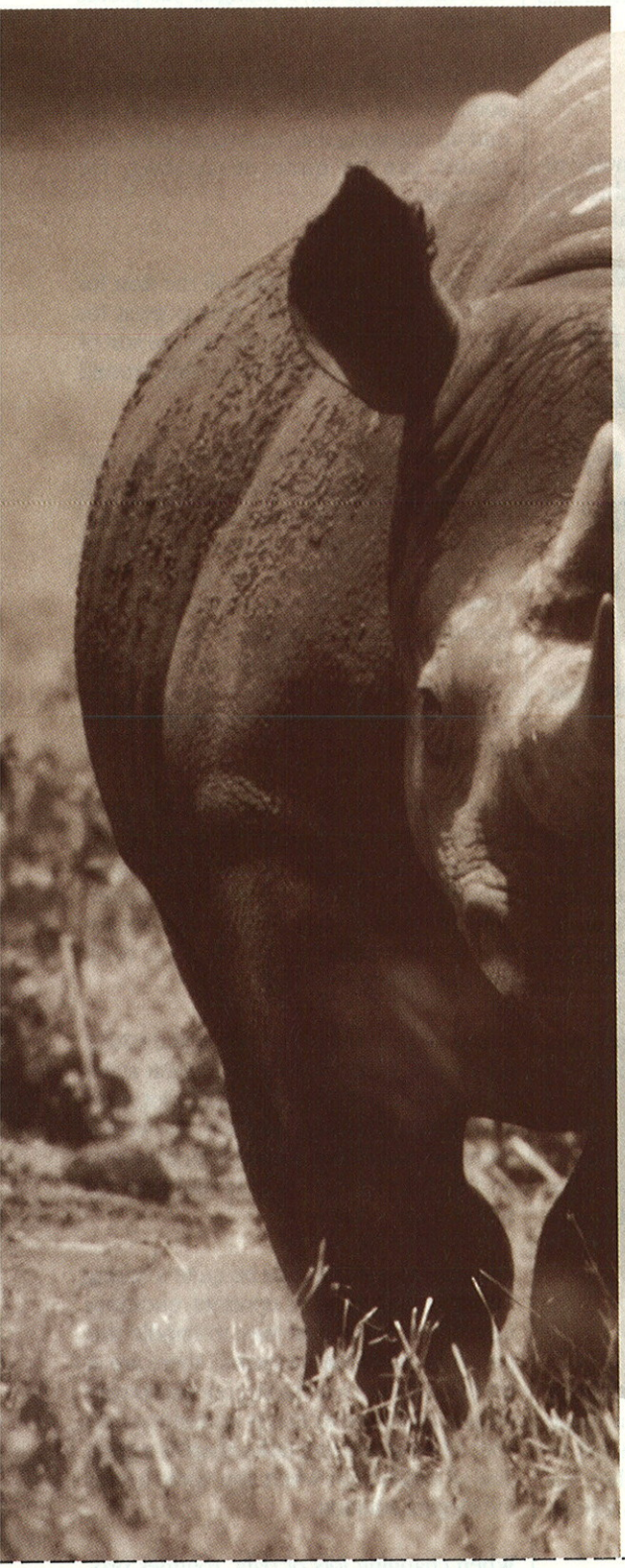
An Ark worksheet looks the same (Figure 2) but it is the position of the labels that determines the location of the data. The concepts - entity names, attributes and others - are placed on the worksheet first and define the meaning of individual data cells. When the concepts are complete and you start to enter data, Ark will only let you fill those cells that have a unique definition.

If you don't include any examples of a particular attribute on the worksheet, Ark creates a context window at the top of the screen. Here you fill in the missing values. By changing the context, you can view other data. For example, the worksheet for the North will have an identical structure to that for the South but with completely different data. To view a different set of data, all you need do is type a different Region name in the context window.

In this way, you need only one worksheet to cover all 60 possible variations, and you can pull out the data for any combination of context instantly.

You can now go on to create other worksheets, based on the same data. This is where Ark really comes into its own. Be-

Software errors can be brutal...
...find them first with Evaluator



Testing Software properly means hours of boredom, and is very, very expensive. But letting users find errors can be as bruising as an encounter with our friend on the left.

Evaluator can help. A unique testing tool, Evaluator can automate the entire testing process, including regression, stress and acceptance testing. It can do this unattended, leaving staff free to perform more productive work.

Evaluator will test every kind of software including GUI applications such as Windows 3. Operating system independent you can test DOS, OS/2 or UNIX applications. If you can get your mainframe application to display on a PC it can test that too!

Evaluator is non-intrusive, which means that the machine you test your software on will be the same as the machine it gets used on; a very important quality consideration.

So if you don't want any unpleasant surprises from your software, call The Products Group for full details on 0285 - 655888

ELVEREX

CIRCLE NO. 070

Tick here for full details:

☐ Evaluator ☐ Consultancy

☐ Product Catalogue ☐ Training: OS/2, UNIX, Windows, C, Object Oriented, Networks.

NAME TITLE

COMPANY

ADDRESS

TEL. NO.

.EXE 8/91

QA

Mail to: The Products Group, QA Training Ltd., Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EF. UK.
Tel: 0285 655888 Fax: 0285 640181

```
FILTER 1990_Actual_North
IF Sales > Target THEN
  INCLUDE Product IN Successful Products
```

Figure 3 - Filling a set

cause of its unique definition, each item of data need be entered only once. For instance, you can keep the product names down the left-hand side but put the Periods along the top. If the word 'Sales' appears in a suitable position on the worksheet, the new context window asks only for Analysis and Region. When you select values for the context, Ark extracts the data and displays it, giving you a whole new way of viewing the data.

Of course, you don't have to stick to one set of data. A single database can cover every aspect of an application, with worksheets containing hundreds or thousands of data items. Once the data is there, you can mix and match to your heart's content.

Tracing the Data

With a database of this size, awash with data, it would be easy to get lost and be overcome by the complexity of it all. This is where the principle of provenance comes in.

Ark remembers where each item of data was originally entered. On any worksheet the data is colour-coded: white for original data, yellow for data entered elsewhere. Move the cursor onto a 'yellow' item and press function key F7, and Ark loads and displays the worksheet where the data was originally entered, with the cursor on the correct cell. This is the provenance of the data.

You can move the cursor to another cell and repeat the process, travelling along a chain of worksheets. At any time, the backtrack key (F6) takes you back a step, until you eventually arrive back where you started.

Programming

So much for the worksheets. The real power of the system only becomes apparent when you start to use the built-in programming language. Programming in Ark is something of a culture shock for those more used to traditional methods.

Once you have sorted out the structure, the next immediate problem is: How do you do anything useful with the data? One answer is to use worksheet formulae. At their simplest these follow broadly the same rules as those of any other spreadsheet. Any cell values are changed with a MAKE statement. For example, to calculate the Total Sales, enter a formula such as:

```
MAKE B85 = SUM(B4..B83)
```

You can have more complex formulae instructions, including a conditional IF...ELSE construct and a REPEAT statement that repeats one or more instructions for any collection of cells.

As far as Ark is concerned it is the definition of the data, not its position, that is important

But suppose you want to test the effects of a 10% price rise by setting the 1992 Prices for Forecast 2 at 10% above those of Forecast 1? Although you could devise a worksheet to do the job, this is an ideal task for an Ark routine.

In most programming languages you would have an instruction in the form:

```
PRICES(3,2,i,j,k) =
 PRICES(3,1,i,j,k) * 1.1
```

This rather unpleasant instruction would be embedded in three nested loops. Ark provides a very neat solution:

```
FILTER Prices_1992
MAKE Forecast 2 =
      Forecast 1 * 110%
```

Within a routine, each MAKE statement acts on a class of data, rather than a single item. The FILTER statement tells Ark to select the class of data that matches the names that follow: ie anything with 'Prices' or '1992' in the tag.

The MAKE statement takes all Forecast 1 data (as restricted by the FILTER) and adds 10%, putting the results in the corresponding Forecast 2 slots. This is repeated for all products and companies. With two instructions you have changed 320 pieces of data.

This is extremely effective but it does need some careful thought, otherwise you can end up changing data unintentionally. The results can be quite devastating; for example:

```
MAKE 1992 = ""
```

This single instruction zaps all the data for 1992 (across all entities). This is great for tidying up your database but not the sort of instruction you want to leave lying around in a routine.

There are some quirks of the system that take a while to get used to. The MAKE statement only acts in those cases where the data already exists on at least one side of the expression. If there are gaps in your Forecast 1 data, then there will be corresponding gaps in Forecast 2. Any existing Forecast 2 data for which there is no Forecast 1 will be deleted. This can be confusing but it is generally solved by good data management.

Sometimes a routine doesn't affect any data and tracking down the cause can be tricky, although Ark has reasonable single-stepping facilities. The answer may lie in generating some dummy data on a worksheet just so that the routine has something to work on!

If you don't restrict the scope of the MAKE statements with a suitable FILTER statement you may end up changing a lot more data than you bargained for. However, there are comprehensive audit facilities that let you list any class of data and find out where it came from.

Programming in Ark can be quite a taxing business and requires constant questioning: Which is the best way to handle this particular problem? Inevitably, there are at least two answers, including one that is very elegant and another that is horrendous! As with all programming tasks, there is a great deal of satisfaction when you come across the elegant solution.

When changing lots of data, routines provide the answer. For complex calculations, it is better to stick to worksheet formulae.

```
CATEGORISE Company IN All Credit Sets
BY Credit Rating + " Companies"
```

Figure 4 - CATEGORISE instruction

You can merge the two in a routine with a DO statement:

```
DO Total Sales
```

This loads the Total Sales worksheet, executes all the formulae and saves away the changes, a far simpler approach than trying to incorporate the formulae themselves in routines. The DO statement is also used to run routines.

To get input from a user, the routine can include a statement such as:

```
INPUT Total Sales
```

This time the worksheet is loaded and the user can make changes to the data (or even the layout). When the user escapes out, the changes are saved and the next instruction in the routine is executed.

In this way, you can set up a worksheet to get any input you like from the user. However, some worksheets may appear somewhat contrived and all are very much in the Ark style. That said, you can create a whole string of bug-free input screens in a matter of minutes.

Sets

As a spreadsheet-type product, Ark is fun to use and in most respects matches all other offerings on the market. But to create powerful applications you need a couple of other features.

The first of these is the set, which is a group of attribute names. For example, the Period attribute may have a set called 'All Years' that comprises 1989, 1990, 1991 etc. Another set, 'All Products', may list the 80 product names. These sets can be used in both worksheets and routines.

To create a worksheet comparing Sales values on a year-by-year basis requires only three concepts: Sales and All Products on the left, All Years at the top. When you change to Data Entry mode (there are several operating modes), Ark does the rest for you. The sets are expanded to show all the names around the edge and the corresponding grid of data is displayed.

Sets can be filled out manually (using the pull-down menus), within a worksheet (by listing them below a set name) or from within a routine (see Figure 3).

As before, these instructions are carried out for all matching data. In this case there are two entities: Sales and Target, defined by the Period, Analysis, Region and Product. For all 1990 Actual data, Ark compares the Sales value with the corresponding Target and, if it is greater, adds the product name to the set of Successful Products. Now all you need to do is type the name 'Successful Products' on a worksheet and Ark will display the required list.

```
WITH This Company = All Companies DO
  DO Create Second Forecast
```

Figure 5 - Representatives in routines

Once you have grasped the principles, you can create complex data structures literally in seconds

If every Company has a Credit Rating (Good, Bad etc), you can create sets such as Good Companies and Bad Companies, where each set contains only the relevant customers. This is achieved with a single CATEGORISE instruction (Figure 4). As an added bonus, you get a set called All Credit Sets that contains a list of the new sets.

Variables

As a programming language, Ark is very unusual in having no variables! There is something similar, though, called a representative. This does what it says - it represents an attribute. For example, This Year may be a representative of Period.

If there is a representative on a worksheet, the user is asked to fill in its value. The corresponding data is loaded. Here, as in many other places within Ark, all you need do is press F5 to select from a pop-up list.

One representative can be derived from another. For instance, the definition of Last Year may be:

```
All Years, This Year - 1
```

If This Year currently has the value '1990', Ark searches the All Years set for 1990 and then puts the previous member of the set in Last Year: 1989, assuming you have ordered the set members correctly.

Put both of these representatives at the top of the worksheet and you need only select one year (This Year) to generate two columns of data.

Representatives are useful in routines. In Figure 5, the routine 'Create Second Forecast' is executed a number of times, each time with the representative (This Company) taking a different value from the set All Companies.

To extend this system further, there are representative sets. These are variables that stand for set names. If there is a representative called New Town, derived from All Towns, then a representative set called New Town's Customers can be derived as:

```
by name, New Town + "Customers"
```

When you enter the name of a New Town (eg 'Hereford'), a Town attribute called Hereford is created and added to the set of All Towns. At the same time the set Hereford Customers is created and can be filled out on a worksheet or within a routine.

This is fairly complex stuff, and takes a lot of getting used to. However, once you have grasped the principles, you can create complex data structures literally in seconds.

Conclusion

Ark is extremely powerful, yet the underlying idea is remarkably simple. You won't master the programming language overnight but, once you have done so, application development time is reduced dramatically. From the end user's point of view the operation of any Ark application is straightforward and requires little training.

Adapting to Ark's way of doing things takes time, especially for a seasoned programmer, but there is no doubt that Ark is going to change the way we all look at and use our data.

EXE

Stephen Morris is a freelance technical author and has written a number of books describing PCs and PC software. He is also the author of the Ark user manuals.
Ark is produced by Ambit Research Ltd, with a starting price of £1950. Ark is available direct from Ambit Research (071 731 8199).

Helping Hands

The Microsoft Windows help system is powerful and slick.
But, as Jeffrey Goldberg discovered, it's not that easy to use in applications.

In a review of the Windows 3 SDK (*.EXE Magazine, November 1990*), I mentioned that Microsoft had provided the tools to allow a sophisticated help system to be added to a Windows 3 program. The tools to create this help system are also now bundled with Borland and Zortech compilers. This article describes how this help system works from a programmer's and help writer's viewpoint.

You might ask: given that the *Programming Tools Manual* of the Windows SDK provides an explanation of the help system, why would someone write an article such as this? Quite simply, the code provided in the manual has a myriad of problems, and the explanations provided are shrouded in Microsoftese. Indeed, the help system looks a last minute addition to the SDK, further confirmed by Microsoft using an older version of the help system for Windows Word 1.1 and Windows Project.

The Windows Help System

The Windows help system provides all the attributes of a good help system and with the important benefit of being a Windows program. It is essential that you know how the Windows help system looks from a user's perspective before delving into the gory details, otherwise you'll get lost; so let's take a closer look.

The context sensitive entrance to the help system is by pressing F1 at a convenient point, for example a menu or in the main window of the application. Unfortunately, this won't give help on subsidiary parts of the application such as the window border or child windows that need to be picked using a mouse. Windows therefore provides the Shift-F1 help mode, which turns the mouse pointer into a combination of a pointer and a question mark, allowing you to move to the feature and press the mouse button to obtain help.

Context sensitivity isn't the only way into the help system. A Windows program should have a help menu as the last item on the menu bar. This allows someone to select the main index, a list of the keyboard short-cuts or procedures, or help on how to use the help system. Although we won't cover Computer Based Tutorials (CBT) in this article because they can't easily be made with the help system, the help menu should contain an entrance to a CBT if one exists. It is inevitable, however, that the Windows help system will move towards a full CBT authoring system in the future.

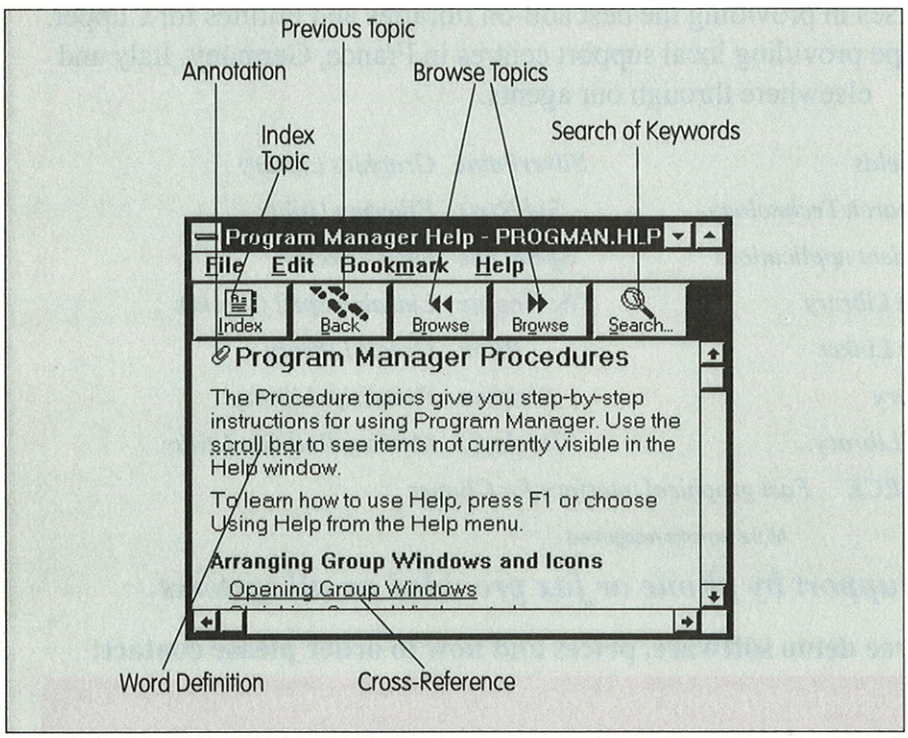


Figure 1 - Screen Dump

As you can see in Figure 1, once in the help system you have many possibilities. A typical topic has normal text interspersed with two special types of words: Cross-References and Word Definitions that you mouse-click on or press Enter, to reveal their contents. A Cross-Reference is a jump to another topic allowing you to find a related topic quickly. For example a Search help topic should have a cross reference to a Replace topic. You are not returned to the previous topic automatically; instead you need to click the Back button. The Word Definition opens up a small window allowing an explanation of a word like an electronic glossary, though you can also use it for a helpful hint on a particular function. When you let go of the button or Enter key, the window closes so you go back to the original topic, at the expense of naïve users who do not know how to keep the definition window open.

The icons at the top of the window make the help system look pretty, while also serving a useful purpose. They allow the user to go to the main index of help topics, to

Everything You Ever Wanted in UNIX for only £99.95*

By the end of this year there will be more people using UNIX™ than any other operating system in the world. But, for many of us UNIX is way beyond our pocket — it might as well be on another planet.

But now Coherent™ gives you everything you ever wanted from UNIX, but for less. You might find that hard to believe, but just consider this.

Coherent is a virtual clone of UNIX, and was developed independently by the Mark Williams Company. That means you don't have to pay hundreds of pounds per copy in licensing fees.

Smaller, Faster and Better!

What's more, Coherent embraces the original principle of UNIX — small is beautiful.

This leads to the fact that Coherent has a whole host of cost and performance advantages. So read on, because there's a lot more to Coherent than just its keen price.

For one thing, Coherent gives you UNIX capabilities on a machine you can actually afford. Needing only 10 megabytes of disk space, Coherent can reside with DOS. So you can keep all your current software and move up to Coherent at your own speed. You can also run it faster, learn it quicker and get better overall performance. All because Coherent is small. Sounds to good to be true, doesn't it?

But small wouldn't be so great if it didn't do the job it was meant to do.

Everything Unix Was Meant To Do and More!

Like the original UNIX, Coherent is a powerful multi-user, multi-tasking development system. With a complete UNIX-compatible kernel which opens up the vast world of UNIX software, including over a gigabyte of public domain software.

Everything You Need for Less!

Coherent also comes with Lex & Yacc, UUCP, a C compiler and a full set of almost 200 UNIX commands for text processing, program development, administration and maintenance.

The latest release includes a UUCP Bulletin Board System, RAM disk support and much, much more.

	Coherent for IBM-PC/AT 286 or 386 machines.	Santa Cruz XENIX 286, Version 2.3.2
No. of Manuals	1	8
No. of Disks	4	21
Kernel Size	64 Kb	198 Kb
Install Time	20 mins	3-4 Hrs
Disk Space	10 Mb	30Mb
Min. Memory	640 Kb	1-2 Mb
Performance*	38.7 sec	100.3 sec
Price	£99.95	£800.00

*Byte Excel benchmark, 1000 iterations on a 20 Mhz 386. Hardware Requirements: 286, 386 or 486 IBM compatible PC with a hard disc & a high density floppy drive. The current release of Coherent does not run on PS1, PS2 or other Microchannel machines, or systems with non-Adaptek 154X series SCSI controllers.

"Mark Williams Co. seems to have mastered the art of illusion: Coherent comes so fully qualified as a UNIX clone, you find yourself thinking 'I can't believe it's not UNIX.'"
UNIX Today! 26 November, 1990.

"... (Coherent) may be the best thing that has happened to UNIX yet."
PC Week, 5 November, 1990.

"I recommend it (Coherent) for anyone wanting to get a feel for UNIX without that sinking feeling in the wallet." Personal Computer World, March 1991.

If your wondering how something as good as Coherent could come from nowhere. Well it didn't. It came from Mark Williams Company, people who've developed C compilers for DEC, Intel, Wang and thousands of professional programmers.

It's Amazing Value!

Yes, we know that £99.95 may be hard to believe. But it's true and we've also made it easy for you to find out for yourself. With our 30-day money-back no-quibble guarantee.

Our No-Quibble Money Back Guarantee

You have to be more than just a little curious about Coherent by now. So why not just do it? Pick up that phone and order NOW. You'll be on your way to having everything you ever wanted in UNIX, and for a lot less than you ever expected.

Call our order lines
021 - 766 8333
021 - 766 5968 (24 Hrs.)
or FAX to 021 - 766 5968

VISA

AET Ltd, Suite 2 Bradford Court,
131 Bradford Street, Birmingham, B12 ONS

*Plus shipping and VAT. Coherent is a trademark of Mark Williams Company. UNIX is a trademark of AT&T. Xenix is a trademark of Microsoft.

CIRCLE NO. 074

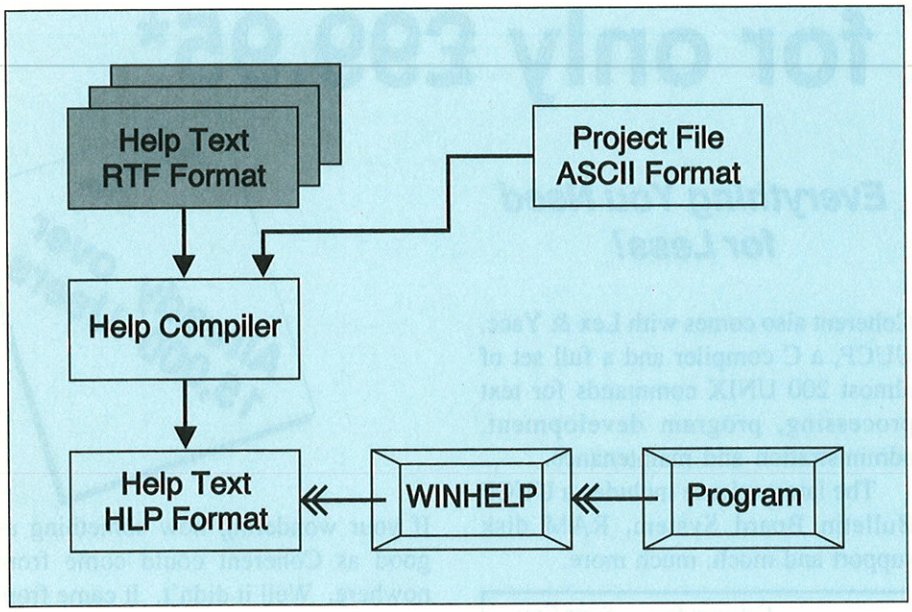


Figure 2 - The Help Compiler.

Value	Meaning and dwData value
HELP_CONTEXT	Shows help for a topic identified by the unsigned integer dwData.
HELP_HELPONHELP	Shows Using Help index topic.
HELP_INDEX	Shows main Help index topic.
HELP_KEY	Shows help for a topic identified by keyword given by the long pointer to a string dwData.
HELP_QUIT	Closes help for your application

Two other settings of wCommand, HELP_SETINDEX and HELP_MULTIKEY, are for when you want more than one index in your help text - a rare occurrence and not covered in this article.

Figure 3 - wCommand values

several previous topics, to browse through related topics and to search for a particular topic. The search isn't a string search, a feature that Microsoft omitted from the help system, but an index search that pops up a dialog box. You type in the Search box or use the list box below to select a Keyword, in the same way that you might search a book index for a particular entry. Once you press the Search button, the help system searches for help topics with that keyword; the ones it finds it puts in the list box at the bottom of the dialog box. The list box contains the titles of the topics related to the index, just like a selection of page numbers related to a single topic in a book index. You then press the Go To button to view the help topic. This is unwieldy, since most help system writers use a one-level system with the same keyword in the index as the title in the topic.

```
case IDM_HELP_INDEX:
  WinHelp(hWnd, szHelpFileName,
          HELP_INDEX, 0L);
  break;

case IDM_HELP_KEYBOARD:
  WinHelp(hWnd, szHelpFileName,
          HELP_KEY,
          (DWORD)(LPSTR)"keys");
  break;

case IDM_HELP_COMMANDS:
  WinHelp(hWnd, szHelpFileName,
          HELP_KEY,
          (DWORD)(LPSTR)"commands");
  break;

case IDM_HELP_PROCEDURES:
  WinHelp(hWnd, szHelpFileName,
          HELP_KEY,
          (DWORD)(LPSTR)"procedures");
  break;

case IDM_USING_HELP:
  WinHelp(hWnd, szHelpFileName,
          HELP_HELPONHELP, 0L);
  break;
```

Figure 4 - Help menu code

The help system also comes with a series of menus to allow you to copy and print a topic, add an annotation and a bookmark to the help topic. The help system only allows you to copy a complete topic so if users are likely to copy a topic frequently (eg a language reference) you need to structure the help text accordingly. The annotation allows someone to add a note to a particular topic - it is entirely user driven with nothing required from help writer or programmer. A bookmark, on the other hand, is different because while the user drops the bookmark, the default title in the resulting dialog box comes from the help topic.

From the user's point of view, the help system is reasonably friendly. Let's see if this is true from the programmer's point of view. The help system consists of additions to the code to support the help system, context sensitivity and the help menu; the help text made with a Rich Text Format (RTF) compatible word processor; and the help compiler's project file. As shown in Figure 2, you feed the help text and the project file into a woefully slow help compiler that generates a compressed .HLP file. The program then uses the Windows API to activate the help application to do the look-up and display of the help topic.

Application Code

The help system has just one innocuous looking API call:

```
BOOL WinHelp (hWnd,
      lpHelpFile,wCommand,
                  dwData)
```

hWnd is a handle to the calling window, allowing the help application to keep track of the applications that request help. The lpHelpFile is the path name of the help file containing the help topic. The wCommand tells the help application the function you want - see Figure 3.

It looks so easy - all you need to do is put calls to WinHelp at appropriate places and let the help writers do their stuff. After you have read the article and perused the source code, it *will* seem that easy, so long as you ignore the Microsoft Programming Tools manual and sample source code.

The first thing you'll need do is to add code that initialises and destroys the help system. Initialising consists of finding where your help file hangs out and constructing a fully qualified pathname for the lpHelpFile parameter of WinHelp. I suggest that you assume that by default the help file is in the

```
/* Somewhere in initialisation code */
hMenu = GetMenu(hWnd);
ahPopMenuID[0] = GetSystemMenu(hWnd, 0);
for (i=1;
     i  GetMenuItemCount(hMenu)+1;
     ++i)
  ahPopUpMenuID[i] =
              GetSubMenu(hMenu,i);

/* In the window procedure */
case WM_MENUSELECT:
  bInMenuBar =
        (BOOL) (lParam & MF_POPUP);
  wCurrentMenuID = wParam;
  break;

case WM_ENTERIDLE:
  if ((wParam == MSGF_MENU) &&
      (GetKeyState(VK_F1) & 0x8000) &&
      !bShiftF1Mode)
    PostMessage(hWnd, WM_HELP,
                wCurrentMenuID, 0L);
  break;
```

Figure 5 - F1 within menus

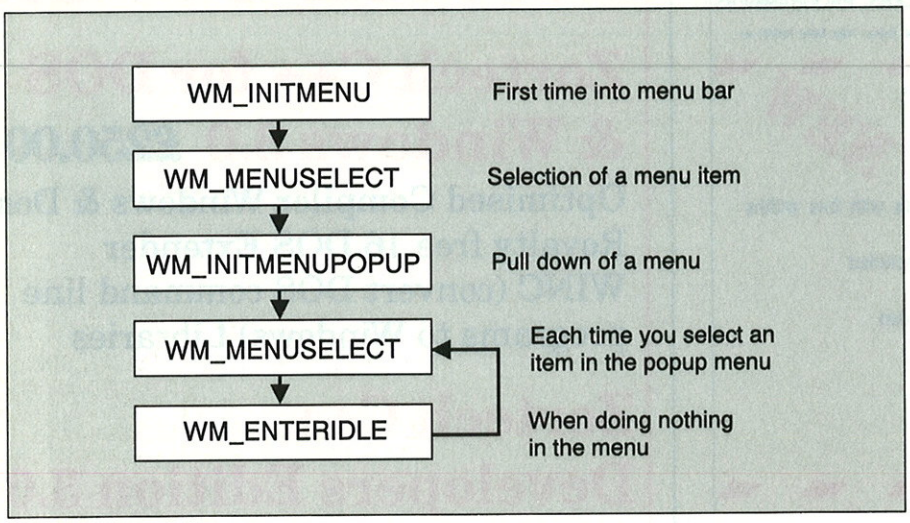


Figure 6 - Menu Messages.

same directory as the executable. You can then find out the path name of the executable with `GetModuleFilename`, remove the filename from the path and substitute the .HLP filename. The help application initialises itself when you first call, but you do need to call `WinHelp` when you're through. To do this you should put a call to `WinHelp`, with a `wCommand` of `WM_QUIT` in your `WM_DESTROY` message handling.

The help menu is the next easiest part. In my worked example, I added a Help menu with a pull-down consisting of Index, Keyboard, Commands, Procedures, Using Help and an About entry leading to a dialog box. The code to handle the menu is within the `WM_COMMAND` handler (Figure 4).

The help text needs to have topics with the keywords keys, commands and procedures defined, otherwise an error will result when someone uses the menu items. You should also add the appropriate menu to the resource file. It is similarly trivial to make pressing F1 within the client area of the application generate help, simply by checking for F1 in the `WM_KEYDOWN` message.

Things start to get tricky if you want someone to be able to press F1 in menus to get help. Windows doesn't provide any hooks for the help system in the menu handler. This means that you have no way of determining the menu id of the selection without adding code to handle the `WM_MENUSELECT` and `WM_ENTERIDLE` messages - please see Figure 5. The first code fragment goes somewhere in your initialisation code, eg `WinMain` or `WM_CREATE`. It gets the handle of the menu bar and then fills an array with the menu ids of the system menu and items in the menu bar. `GetMenuItemCount` gives the number of items in the menu bar and `GetSubMenu` the handle associated with the menu given by position `i`.

The `WM_MENUSELECT` handler gets called for each menu item, so to avoid slowing down menu handling it should do the minimum amount of work. Indeed, all it does is to set a Boolean indicating whether the selection is a pop-up menu and record the current menu ID or pop-up menu handle for later analysis. The `WM_ENTERIDLE` message is produced when Windows detects that you are in a menu or dialog box but doing nothing (see Figure 6). It is safe to do more lengthy here, so this is where we check if someone is in a menu, having pressed F1, and open the appropriate help topic using `PostMessage`. You might think that posting an application defined message `WM_HELP` is wasteful compared to a direct call to `WinHelp`, but it makes for more modular code and allows you to support F1 in dialog boxes.

`WM_HELP` (Figure 7) consists of a very inefficient translation between menu or other ID and help ID; checking a Help ID exists; and then calling `WinHelp` with the right context. You are helped with this translation by the `#defines` beginning with HT (for Hit Test) starting at 0, and the SC (for System Command) starting at 0xF000, so you can place your `IDM #defines` anywhere between 100 and 0xEFFF. You don't have any control over the value of the pull down menu handles, so the check of the array of menu items is only done when in the menu bar.

You now have F1 support in menus and the client area but not in dialog boxes. The

```
case WM_HELP:
  switch (wParam) {

  /* Non-client area item */
    case HTTOPRIGHT:
      dwHelpContextId =
          (DWORD) HELPID_SIZING_BORDER;
      break;

  /* System menu item */
    case SC_MAXIMIZE:
      dwHelpContextId =
          (DWORD) HELPID_MAXIMIZE;
      break;

  /* Pull down menu */
    case IDM_NEW:
      dwHelpContextId =
          (DWORD) HELPID_FILE_NEW;
      break;
    default:
      dwHelpContextId =  (DWORD) 0L;
  }

  /*  Menu Items */
  if (bInMenuBar) {
    dwHelpContextId =
        (wParam == ahPopUpMenuID[0]) ?
            (DWORD) HELPID_SYSTEM_MENU:
        (wParam == ahPopUpMenuID[1]) ?
            (DWORD) HELPID_FILE_MENU  :
        (wParam == ahPopUpMenuID[2]) ?
            (DWORD) HELPID_EDIT_MENU  :
        (wParam == ahPopUpMenuID[3]) ?
            (DWORD) HELPID_HELP_MENU  :
        dwHelpContextId;
  }
  if (!dwHelpContextId)  {
    MessageBox(hWnd,
        "Help not available for item",
        "Help Example",
        MB_OK|MB_ICONINFORMATION);
    return(DefWindowProc(hWnd, message,
                    wParam, lParam));
  }
  WinHelp(hWnd, szHelpFileName,
        HELP_CONTEXT, dwHelpContextId);
  break;
```

Figure 7 - WM_HELP handler

```
int FAR PASCAL HelpKeyFilter
                (nCode, wParam, lParam)
int         nCode;
WORD        wParam;
DWORD       lParam;
{
    LPMSG Msg;
    Msg = (LPMSG) lParam;
    if (nCode == MSGF_DIALOGBOX ) {
      if (Msg->message == WM_KEYDOWN &&
                (Msg->wParam == VK_F1))
        PostMessage(hWnd, WM_HELP,
                    IDM_PASTE, 0L);
    }
    return(DefHookProc(nCode, wParam,
                lParam, &lpfnOldHook));
}

|
|

//Installation and deletion of filter

case WM_CREATE:
  lpfnHelpKeyFilter =
    MakeProcInstance(HelpKeyFilter,
                     hInst);
  lpfnOldHook =
    SetWindowsHook(WH_MSGFILTER,
                   lpfnHelpKeyFilter);
  break;

case WM_DESTROY:
  UnhookWindowsHook(WH_MSGFILTER,
                    lpfnHelpKeyFilter);
  FreeProcInstance(lpfnHelpKeyFilter);
  WinHelp(hWnd, szHelpFileName,
          HELP_QUIT ,0L);
  PostQuitMessage(0);
  break;
```

Figure 8 - F1 dialog box filter

```
case WM_COMMAND:
  if (bShiftF1Mode) {
    bShiftF1Mode = FALSE;
    PostMessage(hWnd, WM_HELP,
                wParam, 0L);
    return (DefWindowProc(hWnd, message,
                          wParam, lParam));
  }
  break;

case WM_KEYDOWN:
  if (wParam == VK_F1) {
    if (GetKeyState(VK_SHIFT)) {
      bShiftF1Mode = TRUE;
      SetCursor(hHelpCursor);
      return
        DefWindowProc(hWnd, message,
                      wParam, lParam);
    }
    else {
      WinHelp(hWnd,szHelpFileName,
              HELP_INDEX,0L);
    }

  }
  else if (wParam ==
          (VK_ESCAPE && bShiftF1Mode)) {
    bShiftF1Mode = FALSE;
    SetCursor((HCURSOR)GetClassWord(
                    hWnd, GCW_HCURSOR));
  }
  break;

case WM_NCLBUTTONDOWN:
  if (bShiftF1Mode &&
                  (wParam != HTMENU)) {
    bShiftF1Mode = FALSE;
    PostMessage(hWnd, WM_HELP,
                wParam, 0L);
  }
  return (DefWindowProc(hWnd, message,
                        wParam, lParam));
  break;

case WM_SETCURSOR:
  if (bShiftF1Mode)
    SetCursor(hHelpCursor);
  else
    return
      (DefWindowProc(hWnd, message,
                     wParam, lParam));

  break;
```

Figure 9 - Shift-F1 dialog support

easiest way is to add a button labelled Help and post an appropriate WM_HELP message. This doesn't allow someone to press F1 and get help, although you can use the short-cut Alt-H. If you really want the ability to press F1 in a dialog box then you need to get into the arcane world of a Windows hook.

Windows provides several hooks to allow an application, or more usually a DLL, to process an event before it gets to an application's message queue. The hook we'll use is WH_MSGFILTER, which calls an application nominated procedure whenever a dialog box, message box or menu has retrieved the message but before it has processed that message. We thus need to filter out the F1 key in a dialog box and post the appropriate WM_HELP message. To install and delete the filter you need to add code to, for example, the WM_CREATE and WM_DESTROY messages (Figure 8).

Shift-F1 support is straightforward compared to supporting F1, for example you can add the code in Figure 9 to your window procedure. The key to Shift-F1 support is the handling of WM_KEYDOWN, which detects Shift-F1 and allows you to escape from the mode. Once in the mode, WM_SETCURSOR makes sure you keep the help cursor when the mouse pointer has made an excursion outside the window. Context detection in Shift-F1 mode uses the WM_NCLBUTTONDOWN message for non-client areas such as the title bar, but for menus we use WM_COMMAND. The latter has the unfortunate side effects such as you can't get help on greyed items, top level menus or on items within the system menu. This isn't a problem, however, as you can get help on these using F1.

The Help Text

I have now shown how to write the code. What about the help text? The help text is a document made into an RTF file with commands as footnotes, underlined, double underlined and hidden text. You can use most word processing attributes in your file including different fonts and font attributes, tabs and paragraph borders. The table in Figure 10 gives a summary of what you can and cannot do.

DOs	DON'TS
Fonts: Tms Rmn, Helv, Courier, Modern, Roman	Fonts: Script and Symbol
Attributes: Font size, Bold, Italic, Underlines	Attributes: Word underline, double underline, superscript, subscript, small caps and strikethrough
Paragraphs (Enter in WinWord)	New Lines (Shift-Enter in WinWord)
Borders and boxes in all attributes	Shaded, absolutely positioned and Side-by-Side paragraphs
Tabs including left, right, centre and decimal	Tables
Paragraph indents and spacing	Right Margin determined by window size
Monochrome and Colour bitmaps	Only monochrome BMP format recommended as colour BMP might not work properly on monochrome screens.
Style Sheets	Footnotes and Field codes
Page Break = Topic End	Document formatting

Figure 10 - DOs and DON'Ts in help text

The centre of the commands is the context string, defined by adding a footnote reference of # and a footnote text of, for example, # HELP_ID_FILE_EXIT. If you want to make a cross-reference or definition to that topic then you use HELP_ID_FILE_EXIT to tell the help compiler the destination. Similarly, in your code when you call WinHelp with HELP_CONTEXT, it is HELP_ID_FILE_EXIT that you use for dwData.

Most help systems allow browsing from one page to another without a help writer's knowledge. However, this isn't what the Windows help system does. Instead, the help writer must specify the topic that precedes and follows the current topic. To specify the sequence you add a footnote reference of + and footnote text of the sequence type and number. For example the file menu could consist of a footnote text of: FileMenu:005, FileMenu:010 and FileMenu:015 and the edit menu: EditMenu:005 etc. If you are on FileMenu:010 then you can go back to FileMenu:005 and forward to FileMenu:015, but not to any EditMenu item. A trap for the unwary is that the help compiler has an ASCII sorting sequence so '15' comes *after* '100', hence the leading zero(s).

Finally, you can incorporate a bitmap into the help text in a number of ways. The simplest is using a Windows word processor to paste the bitmap from the clipboard. However, that becomes wasteful and requires re-pasting when the bitmap changes. The help compiler provides three commands to pull in bitmaps, similar to Field Codes in *Word for Windows*. You enter these commands into the text using curly brackets, for example {bmc ARROW.BMP} treats the bitmap as a character. If you do this, then you can use a bitmap as a cross-reference or definition, allowing the possibility of interesting effects without increasing the size of the help file.

Pulling it together

The connection between the help text, C code and output is the HPJ project file - an ASCII file containing directives like WIN.INI. Figure 11 contains an example.

The FILES section lists the RTF files used to make the help system. The BITMAPS section lists the bitmaps used with the curly

```
[FILES]
helpex.rtf  ; main topics

[BITMAPS]
bullet.bmp

[MAP]
#include <helpids.h>

[OPTIONS]
INDEX=main_index
TITLE=Help Example
COMPRESS=false
ROOT=D:\HELP
```

Figure 11 - Project File

bracket bitmap commands, allowing the compiler to save space by avoiding duplicate copies of the bitmaps. These sections also allow you to produce a `make` file dependent only on the HPJ file, rather than a mess of bitmap and RTF files, because the `FILES` and `BITMAPS` sections list all of the files used in the help system.

```
#define HELPID_EDIT_CLEAR     100
#define HELPID_FILE_EXIT      200
#define HELPID_EDIT_WINDOW    300
```

Figure 12 - MAP header

The `MAP` section is the connection between the code and the help text. It defines the context numbers associated with the help context names - see Figure 12. You include this header file in both the C code and the HPJ file, while the help text contains the context strings such as `HELPID_FILE_EXIT`. This means that each time you add a new context in the program such as a 'File Save all' menu item, you need to add:

1. A `#define`, such as `#define HELPID_FILE_SAVE_ALL 201` in `HELPIDS.H`.
2. An addition to `WM_HELP` to translate `IDM_SAVE_ALL` to the context `HELPID_FILE_SAVE_ALL`.
3. An addition to the help text containing a context string of `HELPID_FILE_SAVE_ALL` and some relevant information on 'File Save all'.

The `OPTIONS` section allows you to tailor the help compiler in many ways, but the example lists only the important options. `INDEX` specifies the help topic shown when someone presses the Index button in the help application. If you don't specify the `INDEX` option then the compiler assumes, usually incorrectly, that the index is the first topic encountered. `TITLE` is the heading given to the help in the title bar of the application. Beware that the title has the word 'Help' automatically appended to the text. The `COMPRESS` option toggles compression of the help file. Keep this `OFF` until you are ready to produce a real version both to speed up the tortoise-like compilation and to allow compression to occur. The reason for the latter is that the help compiler makes a .PH file of repeated phrases but only rebuilds it when none exist. Finally, `ROOT` allows you to specify the directory for bitmaps and RTF files.

EXE

Jeffrey Goldberg likes bungy jumping and is looking for something as exciting in the computer field. Contact him on CIX as Cricket or on 081 455 0801.

Jeff has supplied oodles of example code for this article which we do not have the space to print. However, it is available on disk for free (please see page 1 for details). Please mark your envelopes 'WHELP'.

Introduction

Hypermedia has been talked about for decades, but somehow has never realised its potential. At last the technology we need is in place; now we are poised to start solving the real problems. Here is a hyperlinked document exploring those problems.

Even in the sixties, Ted Nelson was working with the ideas which have now come to be called Hypermedia. After reading his now big Press ↵ to continue

One of the leading organisations for research into Human Factors is SIGCHI, a branch of the ACM. SIGCHI organises conferences every year, dealing with all aspects of investigation into the human Press ↵ to continue

Now that Intel's DVI chips are coming online, and a standard is finally merging for the format of CD ROMs, it would appear that real hypermedia systems are to become a reality at last. Brand new satellites are going up for the express purpose of carrying data in this form, and the global village beckons.

Oh, brave new world, which has such machines in't!

It seems every conference about user-interfaces these days is responding to this technology; people write about how to build casual-use systems for exhibitions, intensive-use systems for research, and even automatic correlation of data inside such systems. Some real projects have been undertaken, most with some degree of success. We have the hardware, we have the software, what more could we want?

This illustrates the real issue with hypermedia; it is not the technology which is important, nor is it the interaction between the technology and the user; the important issue is the information that is encoded into the technology, and it seems nobody is addressing this.

In Shakespeares *"Tempest"*, Miranda is so impressed by the sight of the shipwrecked sailors, (the first men she has ever seen, apart from her father) she exclaims *"Oh, brave, new world, which has such people in't"*. The story operates on several levels; Miranda has an idyllic environment, but the invasion of that environment, and her own natural instincts, conspire to sour it for the rest of the Press ↵ to continue

Aldous Huxley, an incisive and insightful author, treated the role of the individual under repressive governments in a number of works. In later life, he became interested in the benefits of hallucinogenic drugs, particularly in their use in 'non-linearising' perception. In his book, *"Doors of perception"*, Press ↵ to continue

I remember reading a science fiction story some time ago, (which had all the subtlety of plot and character development for which the genre is famous!) which concerned a society in which all the books were written by machine in order to be ideologically sound (an idea stolen, no doubt, from *"1984"*). The people revolted, destroyed the machines, and sat down to write real books of their own. They couldn't - their society was so idyllic they had nothing to write about.

Science Fiction, that branch of literature dealing with future situations, has existed in a recognisable form since Jules Verne. Originally concerning itself with technical achivements, as it matured as a form it became more concerned with social criticism, with the technology acting more as a backdrop to enable situations to develop from the Press ↵ to continue

George Orwell was a committed socialist in his youth, but later turned his back on his previous beliefs, and wrote a number of books exploring the faults which he had previously been unable to see. His most famous work, *"1984"*, approached the position of great Press ↵ to continue

Biography

Jules May spent five years producing and directing animations, mainly for advertising. He has applied advertising techniques to disciplines as diverse as philosophy and embryology. He can be contacted on 0707 44185, or on CIX as jules.

Conclusion

Here are exactly the skills we are looking for. Here are the people who should be building hypermedia systems, but, believe me, they are not going to do it on programmers' salaries.

To be practical on a grand scale, hypermedia products will need to be created by more than just one author explaining his view of the world. Probably there will need to be a programmer close by, and also one other person - the communications specialist. This will be the person who understands what the author wants to convey, and designs the structure and language of the document to explain this most clearly.

It's hard enough to write a book. People who understand their subject clearly enough to be authoritative usually are not very good communicators - and it is very rare for someone who understands communications also to understand a subject well enough to explain it fully. As examples, I'm sure most people have struggled through glutinous textbooks and learned almost nothing, while magazine articles have given a totally clear, but utterly inaccurate picture of the same subject.

Now it is claimed that by allowing an author to stucture his material in forms other than linear we can expect his knowledge will be easier to assimilate. It's not that easy - first, the author must be able to find an appropriate structure (no mean feat in itself!), and then coerce his knowledge into his chosen form, which will be almost impossible if (as is often the case with experts) his knowledge is not highly structured in the first place.

These problems are going to be exacerbated, because the way in which hypermedia is commonly envisioned to work will push it towards the television-style "three-minute culture", leaping from one concept to another at the drop of a cognition.

I remember struggling for four years trying to understand calculus, until a communicatons researcher (who claimed to be a lousy mathematician, by the way) explained it all to me in four hours. There are very few people like that researcher, because the role is a new one.

Advertisers are people who can take a revolutionary product (or an utterly boring one), and in thirty seconds, one hundred words, one picture, or whatever, identify exactly why the product is wonderful beyond dreams. To do this every word, every frame of film, contributes to the message, and by using languages which operate simultaneously on logical, emotional, and subconcious levels, they can cram more information into their allotted space than many authors do in full-length books. They are able to identify isolated, clean ideas and stick to their point throughout their message, and they are so good at this that they can do it when legislation prevents them from saying anything about the product (as in the case of cigarette adverts), or even (as with condoms and tampons) what the product is!

David Ogilvy, one of the most famous advertisers (see "From those nice guys who brought you Pearl Harbor"), regards his subject

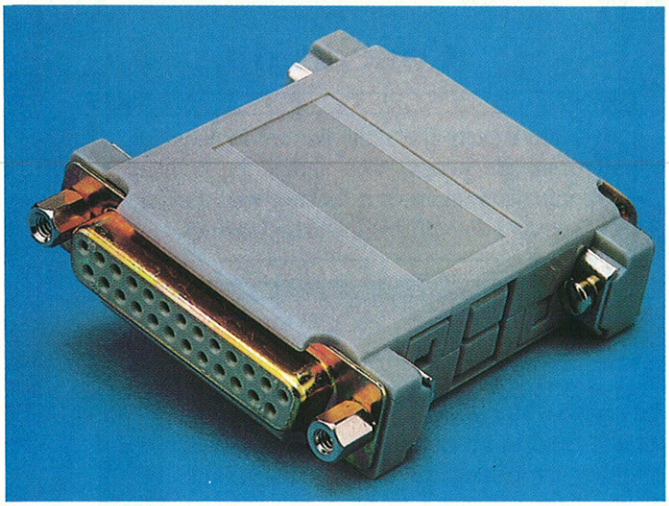
Press ↵ to continue

There are some people at the Open University performing this function, but even the OU is not yet ready for hypermedia, because they insist on calling their specialists "graphic designers" and "scriptwriters", completely missing the point of what these people are doing. Some

Press ↵ to continue

Max Headroom alluded incidentally to the technology and drugs now known as cyberpunk. It described a synthetic TV show which has evol-

Press ↵ to continue



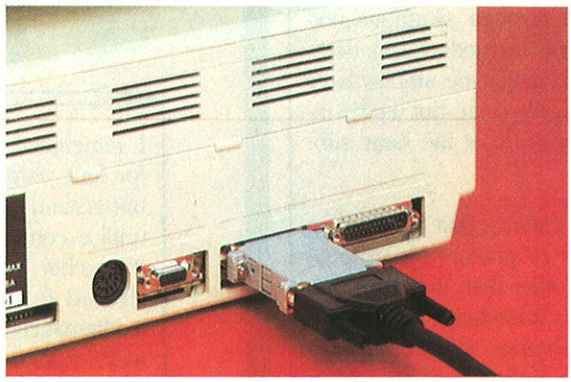
How many users of your software paid for it???

SOFTLoK International Limited was established in 1987 with the introduction of our SOFTLoK and SOFTLoK PLUS devices to combat the ever increasing problem of software piracy. Our SOFTLoK range of software protection devices are used by hundreds of software developers from small consultancies to large multinationals. To cope with the ever changing needs of our customers we have developed SOFTLoK II which combines the programmable features of SOFTLoK PLUS with a low unit cost similar to our original SOFTLoK product.

SOFTLoK II™

The Next Generation

SOFTLoK II units are programmable devices containing read/write memory protected by a password. Both the memory and the password can be changed at any time using our routines in your application software. Easy to use menu-driven software is provided to allow small or large batches of SOFTLoK II units to be programmed with their initial data & passwords ready to be sent out with the protected software product.



As SOFTLoK II units plug into the parallel printer port they can be installed or removed in seconds.

Price: 1-19 £16.90, 20-49 £15.40, 50+ £13.70

Evaluation kit £20 (SOFTLoK II, manual & software)
All prices excl. VAT and delivery

- For IBM PC, PS/2 and compatibles
- Uses parallel printer port
- Totally transparent to printer
- Secure data & password can be changed from your application software
- Cascadeable
- 240 bytes of secure read/write memory
- 8 byte (64 bit) password
- No programming adaptors required
- Easy to use SOFTLoK II setup software
- Routines ready to link with various compilers
- Easy to follow manual

I-MEX House, 40 Princess Street,
Manchester, M1 6DE, England,
Tel: 061 228 7379 **Fax:** 061 236 6890

CIRCLE NO. 082

A load of old COBOL

COBOL is hardly an obscure language, although some say it should be. Christina Wheeler finds some strengths among its weaknesses.

The first version of COBOL was produced by the Conference on Data Systems Languages (CODASYL) in 1959. The requirements of the design were that it should be easy to understand and document, easy to convert from one machine to another, and capable of being compiled into efficient object code for data processing problems.

The American National Standard Institute approved a standard version in 1968. However, the standard was not really adopted until it was updated in 1974. Most COBOL in use today is based on the 1974 standard, ANS COBOL 74, although most manufacturers also add their own extensions.

Another version, ANS COBOL 85, was released to provide better facilities in the language for structured programming. This included a 'case' statement (EVALUATE) and an ENDIF. Unfortunately, this version is still rarely used.

The language

COBOL was originally designed to look like English, in order that business people could write their own programs without recourse to computer professionals. No comment. Its basic building block is a word; either a reserved word (such as READ, ADD, PERFORM etc) or a user-defined word, which can have up to 30 characters. Admittedly, very few programmers would use all thirty! Even so, the language seems very long winded when you come to it from another language.

Figure 1 gives an idea of the structure of a COBOL program. Each DIVISION consists of SECTIONs. These have fixed names for the first three, and user-defined names for the PROCEDURE DIVISION. SECTIONs may be further divided into PARAGRAPHs. Each of these is made up of a series of sentences, composed of clauses. This terminology clearly reflects the attempt to make the language English-like.

For all its faults, COBOL does have some strengths, notably the ability to describe business data. This is, of course, what it was designed for in the first place. What amazes me is that it continues to be used for applications for which it is not really suitable, and therefore ends up being unnecessarily maligned. Which brings us to the Triangle Problem. For such a small program, COBOL is indeed top heavy, as you still need to include all four divisions. Figure 2 shows my solution to the problem.

The IDENTIFICATION DIVISION

This identifies the program! As well as the name of the program, other information may be included, such as author's name, the name of the installation and so on. A comment about the function of the program as a whole is usually included here.

The ENVIRONMENT DIVISION

The CONFIGURATION SECTION gives the types of computer on which the program is to be compiled and run. Portability was another of COBOL's original design considerations. Maybe my cynicism in this area is ill-founded, because I actually ran this program on a PC.

If the program were handling files, there would also be an INPUT-OUTPUT SECTION in the ENVIRONMENT DIVISION, connecting the program's internal file names to those used by the operating system.

The DATA DIVISION

The layout, as perceived by the program, of any such files would be described by the FILE SECTION of the DATA DIVISION. Each file is described with its format and the layout of however many types of record are required. COBOL is best suited to batch file processing and additional software may be brought into play for on-line processing or handling databases.

Working areas are described in the WORKING-STORAGE SECTION. The descriptions for WORKING-STORAGE and FILEs are much the same. Elements can be grouped into logical chunks and a hierarchy described using level numbers. Thus,

```
IDENTIFICATION DIVISION.

     I think I'll bake a cake!

ENVIRONMENT DIVISION.

     I'll make it in the kitchen.
     We'll eat it in the sitting-room.

DATA DIVISION.

FILE SECTION.

     I'll need Eggs, Flour, Butter, Sugar, Milk, Fruit and Spices
     to go in it.
     When it's done, we'll have fruit cake for tea.

WORKING-STORAGE SECTION.

     I'll need a bowl to mix it in, and a wooden spoon to mix it
with.
     I'll need a tin to bake it in and a hot oven to bake it.
     I'll need a rack to cool it on and a plate to serve it on.

PROCEDURE DIVISION.

MIX  SECTION.

     Get the eggs, butter and milk from the fridge.
     Get the flour, sugar and spices from the cupboard.
     Put them in the bowl and mix them with the wooden spoon.

BAKE SECTION.

     Transfer the mixture to the tin.
     Put it in the hot oven.
     Leave it there until it is cooked.

COOL SECTION.

     Remove the tin from the oven.
     Remove the cake from the tin, and put it on the rack to cool.

TEA  SECTION.

     When it is cool, put it on the plate and take it to the
     sitting-room for tea.
```

Figure 1 - A light-hearted look at the structure of COBOL

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. TRIANGLE.

      ******************************************
      *                                        *
      * COBOL SOLUTION TO THE TRIANGLE PROBLEM *
      *                                        *
      ******************************************

       ENVIRONMENT DIVISION.

       CONFIGURATION SECTION.
       SOURCE-COMPUTER. IBM-370.
       OBJECT-COMPUTER. IBM-370.

       DATA DIVISION.
       WORKING-STORAGE SECTION.

      *  ARRAY TO HOLD THREE INPUT VALUES.
       01  TRIANGLE-SIDES.
           03  LENGTH-OF-SIDE   PIC 9(9)  COMP  OCCURS 3.

      *  NUMBER OF SIDES (CONSTANT)
       77  NO-OF-SIDES          PIC 9 VALUE 3.

      *  SUBSCRIPT FOR ARRAY.
       77  L-SUB                PIC 9.

       PROCEDURE DIVISION.

       MAIN-PROCESS SECTION.
      *  CONTROLLING LOOPS.

           DISPLAY "ENTER THREE LENGTHS".
           PERFORM READ-NUMBER
               VARYING L-SUB FROM 1 BY 1
                   UNTIL L-SUB GREATER THAN NO-OF-SIDES.
           PERFORM PRINT-NUMBER
               VARYING L-SUB FROM 1 BY 1
                   UNTIL L-SUB GREATER THAN NO-OF-SIDES.
           PERFORM CHECK-VALUES.
           STOP RUN.

       READ-NUMBER SECTION.
      * READ A NUMBER INTO THE NEXT ARRAY ELEMENT.
           ACCEPT LENGTH-OF-SIDE (L-SUB).

       PRINT-NUMBER SECTION.
      * PRINT THE NEXT ARRAY ELEMENT.
           DISPLAY "LENGTH " L-SUB " = " LENGTH-OF-SIDE (L-SUB).

       CHECK-VALUES SECTION.
      * DETERMINE TYPE OF TRIANGLE (IF ANY).
           IF LENGTH-OF-SIDE (1) GREATER THAN
              (LENGTH-OF-SIDE (2) + LENGTH-OF-SIDE (3))
           OR LENGTH-OF-SIDE (2) GREATER THAN
              (LENGTH-OF-SIDE (1) + LENGTH-OF-SIDE (3))
           OR LENGTH-OF-SIDE (3) GREATER THAN
              (LENGTH-OF-SIDE (1) + LENGTH-OF-SIDE (2))
               DISPLAY "THIS IS NOT A TRIANGLE"
           ELSE
               IF  LENGTH-OF-SIDE (1) = LENGTH-OF-SIDE (2)
               AND LENGTH-OF-SIDE (1) = LENGTH-OF-SIDE (3)
                   DISPLAY "THIS IS AN EQUILATERAL TRIANGLE"
               ELSE
                   IF LENGTH-OF-SIDE (1) = LENGTH-OF-SIDE (2)
                   OR LENGTH-OF-SIDE (1) = LENGTH-OF-SIDE (3)
                   OR LENGTH-OF-SIDE (2) = LENGTH-OF-SIDE (3)
                       DISPLAY "THIS IS AN ISOSCELES TRIANGLE"
                   ELSE
                       DISPLAY "THIS IS A SCALENE TRIANGLE".

      * END OF PROGRAM TRIANGLE
```

Figure 2 - COBOL solution to the triangle problem

LENGTH-OF-SIDE is an element within TRIANGLE-SIDES. Of course, in most cases, there would be more than one element at the lower level.

The OCCURS clause specifies that there are three elements called LENGTH-OF-SIDE making up the array TRIANGLE-SIDES. I have chosen to access these using a subscript, defined later as L-SUB. It is also possible to use an index that would be defined with the array, would have a format determined by the operating system, and would allow a number of additional functions, notably a SEARCH command.

The format of the elements is determined by the PIC clause (short for PICTURE) - in this case, nine digits. COMP (short for COMPUTATIONAL) specifies a binary field. Other forms of COMPUTATIONAL field may be available, depending on the machine. For example, in IBM mainframe COBOL, COMP-3 would specify a packed field. If COMP is omitted, the field is 'display numeric' (ASCII or EBCDIC).

The combination of level numbers and PICTURE clauses allows great flexibility in describing data structures. The higher level items, known as 'group' items, can be referred to as one block. Only items at the lowest level can have a PICTURE. Group items are considered to be alphanumeric by definition. Thus, TRIANGLE-SIDES is an alphanumeric field. You could actually assign an alpha value to it, with all the attendant fun and games, if you then tried to do anything with the numeric array elements. Of course, this facility can also be turned to an advantage.

Special facilities

There are certain level numbers which have specific meanings. I have used one here, 77, which is used for odd fields which are not part of a data structure. However, on some machines, it is more memory efficient to group all the odd data items together under an 01 level (often called ODDS-AND-SODS!).

The special level I have found most useful is 88, used to describe Boolean variables. The value of the 88-level element can be either 'TRUE' or 'FALSE', depending on the value of an associated field. These can appear anywhere in a data structure. Look at Figure 3. Assume some piece of processing is dependent on TYPE-OF-DATA having a value of 1, 3 or 5. Instead of a long winded IF statement with a couple of ORs, the statement can simply read 'IF ALLOWED-DATA-TYPE...'. This feature was designed primarily to improve readability, but I have also found it a lot easier to maintain a list of values in the DATA DIVISION than to rummage around in the code of the PROCEDURE DIVISION.

```
01  SOME-DATA.
    03  TYPE-OF-DATA          PIC 9.
    88  ALLOWED-DATA-TYPE     VALUES 1 3 5.
    03  NEXT-ITEM-OF-DATA     PIC 9(9).
    ...
```

Figure 3 - A Boolean variable

All data used by the program must be described in the DATA DIVISION. The data types cannot be changed during execution, and most compilers are quite hot on not allowing invalid processing (eg arithmetic on alpha fields).

Back at the problem

The PROCEDURE DIVISION

Let's get back to the current problem. For input and output, I have chosen to use DISPLAY and ACCEPT, which can be used to communicate directly with a terminal.

DISPLAY simply writes a string of literals and data elements to the screen. It is very unstructured, for example leading zeros cannot be suppressed. Other commands may be available, depending on the machine. IBM COBOL, for instance, has a verb 'EXHIBIT', which allows a more formatted layout of the values to be written.

ACCEPT reads a value from the screen into the specified data element.

The main command for structuring COBOL programs is PERFORM. This causes a SECTION to be processed. A SECTION lasts from one heading (eg PRINT-NUMBER SECTION) to the next (eg CHECK-VALUES SECTION). If you leave off the word SECTION, the name is assumed to be a paragraph heading, or label, and processing continues merrily on. This can be wonderfully difficult to spot. It is also possible

THE SOFTWARE CONSTRUCTION COMPANY

BORLAND

Borland C++	£199
Turbo C++ 2nd. Edition	£55
Turbo Debugger & Tools	£85
Turbo Pascal v.6.0	£69
Pascal Professional	£129
Pascal for Windows	£99
ObjectVision	£79
Paradox 3.5	£335
Paradox Engine	£199
Paradox SQL Link	£215

AUTHORISED

DATABASES

Advanced Revelation	£795
Advanced Revelation -	£695
askSam	£255
askSam LAN	£579
Clarion Personal Dev. 2.0	£79
Clarion Professional Dev.	£679
DataEase	£485
DataEase - LAN 3-PAK	£485
Dataflex	£449
Dataflex Multiuser	£1065
Knowledgeman/2	£645
Magic PC	£325
Open Access III	£449
Open Access III Compiler	£385
R:BASE 3.1	£515
R:BASE Compiler	£190
R:BASE 3.0 LAN Pack	£645

DBASE

Clipper 5.0	£325
dBASE III PLUS	£395
dBASE IV	£495
dBASE IV Developer's Ed.	£695
dBFast/DOS	£255
dBFast/PLUS	£255
dBXL	£165
dBXL LAN	£389
Force	£449
FoxBASE+	£295
FoxBASE+/386	£425
FoxPro	£515
Vulcan	£320

OBJECT-ORIENTATED

Smalltalk/V	£65
Smalltalk/V 286	£129
Smalltalk/V Windows	£325

CASE TOOLS

EasyCASE Plus	£190
EasyCASE Plus Prof.	£255
EasyCASE Prof. Upgrade	£80
Visible Analyst	£385
Visible Analyst Workbench	£2460
Visible Analyst Workbench	£6995
Visible Analyst Workbench	£1295
Visible Dictionary	£595
Visible Prototyper	£595
Visible Rules	£595

VERSION CONTROL

MKS RCS 5.1	£165
MKS RCS 5.1 - 5 User	£645
PVCS Professional	£319
PVCS - 5 User	£1345
SMS	£395
Sourcerer's Apprentice	£175
Sourcerer's Apprentice	£429

TEXT AND UNIX TOOLS

MKS AWK	£65
MKS AWK DOS & OS/2	£129
MKS Programmer's Platform	£429
MKS Toolkit	£165
PolyAWK	£129
PolyAWK Toolkit	£145
SpellCode	£65

TRANSLATORS

BASTOC	£515
BAS_C Commercial	£579
dBx/dBPort	£389
dBx/dBPort with source	£645
FOR_C	£359
Pascal to C Translator	£99

Microsoft

BASIC PDS 7.1	£219
Visual BASIC	£99
C 6.0 Compiler	£210
C 6.0 & Windows SDK	£349
COBOL 4.0	£345
FORTRAN 5.1	£199
MASM 6.0	£65
MS-DOS 5 Upgrade	£59
Pascal Compiler	£149
QuickBASIC	£50
QuickC	£50
QuickC/Assembler	£85
QuickPascal	£50
Source Profiler	£49
Windows 3.0	£65
Windows S.D.K.	£260

Authorised

EDITORS

BRIEF for DOS & OS/2	£199
BRIEF upgrade	£60
BRIEF with dBRIEF	£275
BTags	£35
Cheetah	£159
dBRIEF for dBASE/Clipper	£99
dBRIEF for Paradox	£99
dBRIEF for R:BASE	£99
EDT+	£190
EMACS	£210
Epsilon	£129
KEDIT	£99
MKS Vi	£99
Multi Edit Professional	£115
Norton Editor	£50
SPF/PC	£159
SYNDIE	£319
VEDIT PLUS	£120

LINKERS

.RTLink	£190
.RTLink/Plus	£319
ALINK	£129
Blinker	£175
Plink w/LTO	£359

MAKE UTILITIES

MKS Make	£99
MKS Make - 5 User	£320
PolyMake	£129
PolyMake - 5 User	£485

PARSERS AND LINTS

MKS Lex & Yacc	£165
PC-Lint	£90

PROFILERS

.RTLink/Plus	£319
CHARGE	£65
Codesifter	£79
Inside!	£85
PC Metric	£129

DEMO BUILDERS

Dan Bricklin's Demo II	£190
Instant Replay III	£99
Instant Replay Prof.	£385
Show Partner F/X	£259

MEMORY MANAGERS

386 MAX Version 5	£85
Above Disk	£79
Blue Max	£100
HEADROOM	£85
HEADROOM Network Extens	£39
HI386 Complete	£65
MOVE'EM	£59
QEMM-386	£65
QEMM-50/60	£65
QRAM	£55
Turbo EMS	£65

DOCUMENTATORS

4c for Brief	£79
ABC Flowcharter	£190
ASMFLOW	£129
C-Cleanly	£85
C-Doc	£120
Clear for C	£129
Clear for dBASE	£129
dAnalyst for C and C++	£190
dAnalyst for Clipper	£190
dAnalyst for dBASE III	£190
dAnalyst for dBASE IV	£190
dAnalyst for FoxBASE +	£190

FREE
Microsoft Flight Simulator
WORTH £45 WITH EVERY PRE-PAID ORDER OVER £100*

EasyFlow	£99
Flow Charting III	£165
SourceDoc	£190
The Documentor	£190
Tree Diagrammer	£65

DISASSEMBLERS

Sourcer 486	£85
Sourcer w/BIOS pre-proc.	£110

E-MAIL

cc:Mail	£450
cc:Mail EXPAND	£385
cc:Mail Remote	£190
DaVinci eMAIL	£645
DaVinci eMAIL Combo	£965

CODE GENERATORS

DATABOSS	£395
Matrix Layout	£195
ObjectVision	£259
ObjectVision for Windows	£259
PRO-C	£259
R&R Code Generator	£99
UI2 Developer's Release	£385
UI2 Touch & Go	£259
Zachary	£259

DEBUGGERS

Periscope 286/386 Pod	£389
Periscope I OK	£319
Periscope I/512K	£385
Periscope I/MC	£449
Periscope I/MC 512K	£449
Periscope II	£145
Periscope II-X	£129
Periscope IIw/Switch	£145
Periscope IV (16 MHz)	£775
Periscope IV (25 MHz)	£900
Sherlock	£129
SVS DBG	£210

MULTI-TASKING

DESQview 2.2	£85
DESQview 386	£125
Dr. Switch-OnCall	£65

BATCH ENHANCERS

Batcom	£39
Beyond.BAT	£65
Command Plus	£85
ProKey Plus	£85
The Builder	£99

TERMINAL EMULATORS

BLAST II	£195
COTERM/100	£145
COTERM/220	£160
COTERM/220	£160
COTERM/4010	£165
EM 320	£199
EM 4010	£230
EM 4105	£330
EM 4105 Network	£345
ICE.10	£190
SmarTerm 125	£190
SmarTerm 2392	£65
SmarTerm 240	£225
SmarTerm 320	£129
SmarTerm 340	£225
SmarTerm 400	£115
SmarTerm 4014	£145
Smarterm 470	£190
ZSTEM 100	£65
ZSTEM 220	£99

PROFESSIONAL SHAREWARE FOR DEVELOPERS

ASM Library (20)	£40
C/C++ Library (68)	£66
dBASE/Clipper (87)	£100
DOS Utilities (37)	£40
Novell Netware (54)	£66
Turbo Pascal (38)	£40
Windows (64)	£66

(Number of diskettes)

*All prices exclude V.A.T. U.K. carriage included. ***One offer at time of purchase, per order at published prices.*** VISA or ACCESS accepted.
Credit to approved accounts.
SUMMER '91 CATALOGUE
Call for your free copy of our 160 page catalogue, TOOLBOX.

1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB
TELEPHONE (0763) 244114 FACSIMILE (0763) 244025

CIRCLE NO. 083

to PERFORM paragraphs quite legitimately, which further adds to the excitement.

PERFORM ... VARYING... allows one or more variables to be altered at each iteration until a condition becomes true. The condition is tested at the start of the loop, so that a loop may not be processed at all if the condition is true at the beginning. Any variables are incremented (or indeed decremented) after the relevant SECTION has been processed.

Not perfect after all

To my mind, the single major design fault in ANS COBOL 74 is the lack of a terminator for a conditional block of code. Look at the IF statement in Figure 2. It begins with IF, and ends, right at the very end, with a single full stop. Suppose I wanted to print out 'END OF CHECK-VALUES' before terminating the processing, but only if the triangle was valid. My sole option is to repeat that piece of code within every clause of the IF statement to find a valid triangle, as shown in Figure 4. This is not too bad if there is only one statement. If there is more, it can be infuriating. In practice, long blocks of code would be transferred to separate SECTIONs and replaced by a single PERFORM. Nevertheless, each of the relevant SECTIONs would have to contain that repeated code.

```
CHECK-VALUES SECTION.
* DETERMINE TYPE OF TRIANGLE (IF ANY).
    IF LENGTH-OF-SIDE (1) GREATER THAN
       (LENGTH-OF-SIDE (2) + LENGTH-OF-SIDE (3))
    OR LENGTH-OF-SIDE (2) GREATER THAN
       (LENGTH-OF-SIDE (1) + LENGTH-OF-SIDE (3))
    OR LENGTH-OF-SIDE (3) GREATER THAN
       (LENGTH-OF-SIDE (1) + LENGTH-OF-SIDE (2))
        DISPLAY "THIS IS NOT A TRIANGLE"
    ELSE
        IF  LENGTH-OF-SIDE (1) = LENGTH-OF-SIDE (2)
        AND LENGTH-OF-SIDE (1) = LENGTH-OF-SIDE (3)
            DISPLAY "THIS IS AN EQUILATERAL TRIANGLE"
            DISPLAY "END OF CHECK-VALUES"
        ELSE
            IF LENGTH-OF-SIDE (1) = LENGTH-OF-SIDE (2)
            OR LENGTH-OF-SIDE (1) = LENGTH-OF-SIDE (3)
            OR LENGTH-OF-SIDE (2) = LENGTH-OF-SIDE (3)
                DISPLAY "THIS IS AN ISOSCELES TRIANGLE"
                DISPLAY "END OF CHECK-VALUES"
        ELSE
                DISPLAY "THIS IS A SCALENE TRIANGLE"
                DISPLAY "END OF CHECK-VALUES".
```

Figure 4 - An alternative IF statement for program TRIANGLE

Conclusion

A remark by one of my university lecturers concerning FORTRAN could also be applied to COBOL: 'It has grown like a weed'. I do remember finding COBOL horribly long-winded and rather archaic when I began using it. Like many another, I got used to it. Also, as with most languages I have encountered, it is fine if you use it for what it was designed for. The question as to why languages are used for inappropriate applications might make a good subject for a doctoral thesis.

EXE

After doing a degree in French and Computer Science, Christina Wheeler worked for seven years as a COBOL programmer, mostly in maintenance and support. For the last five years, she has been doing other DP related work, including training poor unsuspecting graduates in the mysteries of COBOL.

STOP SOFTWARE PIRACY

ZipitUP™

Totally Protects MSDOS Programs from Piracy, Illicit Copying and Unauthorised Use

£25 Evaluation Pack Call for details

* Easy to use - No Programming - Software Only System
Versatile & Transparent, runs on *all* 3.5"/5.25" LD/HD Formats
* Use *Zipit*UP on your PC, PC Disk Autoloader or you can bulk duplicate

Stop it happening to Your Software - Contact us **NOW!!!**
Tel: (0865) 842224 Fax: (0865) 841507

THE **data** BUSINESS Experts in Data Resources
3 Bankside, Kidlington, Oxford OX3 1JE

*Zipit*UP - the Best Answer There Is. . .

CIRCLE NO. 085

F77L-EM/32

Port 4GB mainframe applications to 80386s with this 32-bit DOS-Extender compiler. The winner of PC Magazines's 1988 Technical Excellence Award just got better. New Version 3.0 and OS include: Editor, Make Utility, Virtual Memory Support, DESQview Support, New Documentation and Free Unlimited Runtime Licenses. F77L-EM/32 and OS/386. POA.

F77L

The compiler of choice among reviewers and professionals. Includes a Debugger, Editor, Profiler, Linker, Make Utility, Weitek and 386 Real-Mode Support, Graphics. POA.

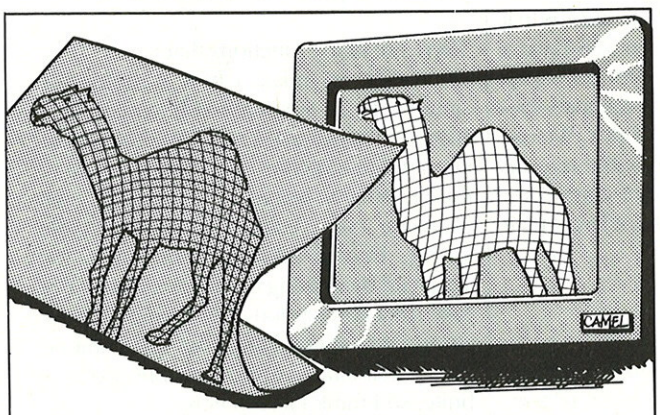
Lahey Personal Fortran 77

New Version 3.0: Full ANSI 77, Debugger, Editor, Linker, Library Manager, Microsoft and Borland C interfaces, 400 page Manual. POA.

System Science

3-5 Cynthia Street,
London N1 9JF
Tel: (071) 833 1022
Fax: 081 837 6411

CIRCLE NO. 103



PLOT THE DIFFERENCE

PLOTVIEW

Is the simple, low cost alternative to paper plots. Able to preview plot images on a graphics screen within seconds.

PLOTVIEW allows paperless plotting and replotting – avoiding delays, improving efficiency.

PLOTVIEW

Has up to 25 separate plot files previewable simultaneously on screen and a zoom function which allows any part of the plot to be inspected in detail.

Easy to integrate, PLOTVIEW is ideal for system builders and can be customised to suit their individual needs.

Phone now for more details about PLOTVIEW: the versatile tool that can really make a difference to your productivity.

CAMEL SERVICES LTD.

Telephone OXFORD (0865) 512678

CIRCLE NO. 086

RM /Cobol-85 RRP £1050.00

- Ver 5 adds features incl. pop-up windows
- GSA certified ANSI Cobol-85 compiler
- Compiles to portable object code.
- Available for DOS, OS/2, AIX, Unix, Xenix.

RM /Panels RRP £330.00

- Forms design and Screen Managment system
- WYSIWYG Rapid screen prototyping

RM /CO* RRP £170.00

- Application development environment
- Edit, compile debug and project management.

Special Offer

RM Cobol-85, RM/Panels and RM/CO* for DOS at our very special price of £995.00 plus VAT, only from System Science

Prices are exclusive of VAT, but include UK delivery

System Science

3-5 Cynthia St, London N1-9JF

Tel: (071) 833-1022 Fax: (071) 837-6411

CIRCLE NO. 087

A little Hashing

Hashing is a simple and fast method for storing and retrieving data. Ray Jones shows how it's done.

Applications requiring the storage and subsequent retrieval of data are countless; from the symbol tables generated by assemblers and compilers for keeping track of variable names, through simple card files and directories, to full-blown databases.

Small stores, consisting of only a handful of data, can be built and searched sequentially without serious impact on the efficiency of the application. However, as table sizes get bigger, more sophisticated methods must be employed. This is in order to maintain reasonable response times for database queries and to ensure fast compile or assembly times.

The use of Hash Tables is one such method. Hashing techniques provide a way of quickly storing and retrieving data records. The method may be used for both large and small applications; the one that I shall present here is quite simple, although the technique is useful for many, more serious, applications.

As all database aficionados know, data records are usually stored and retrieved by means of a key field. In the case of a compiler symbol table this would be the symbol's name; other data would be stored along with this, perhaps the symbol's address and its type. It would be a neat trick, then, if the key field could be translated directly into an index which would allow direct access to the data in an array. This could save a great deal of time and effort in searching for the data.

Furthermore, given that the key field is unique, (and it should be - two variables with the same name and in the same symbol table would not be very useful) this is theoretically possible.

Index problems

Using the compiler symbol table as an example once more; each symbolic name (represented as a string) could be treated as an integer. The trouble is that a variable called, say, 'state table' represents rather a large (11 byte) integer. The table required to accommodate such an index would need to be phenomenally big (around 3 million elements) and the redundancy, for most practical purposes, would be ridiculous. This example is, of course, ludicrously naïve, but the point remains that the formulation of a unique index of a reasonable size from an arbitrary string is not practical.

But a non-unique index will do. What if we give up trying to find a unique index from our string and make do with one that is not quite so special? Here's the plan: we have an array that is big enough to hold all the records that we will need to store, and we also have a formula into which we put the key field string and which will give us a pretty-damned-special (but not actually unique) index. We use our index to place our data into the array, or, of course, to retrieve it. And of course, we must ensure that the index is not larger than the size of the array.

What it's about

To illustrate this idea I'm going to develop a simple database program: a personal telephone directory. This is not intended to be a serious application, but could be tailored to be part of something larger and more sophisticated. The program will have two functions: the first, to allow a name and number to be added to the directory; and the second to display the number of a specified name that is contained in the directory. Neither name nor number will have any special format, they will be simple strings. Since we have to supply a number to match a name (and not vice versa) the name string will be the key field.

Sorry, wrong number

As I mentioned before, the index generated for any name will not be unique. So what happens if the same index is generated more than once? For example, the string "Ray Jones", is an anagram of "Jason Rey" (which is a name that I must bear in mind if I ever resort to writing romantic fiction). So any function used to produce an index that takes only the single characters of the string into account (eg XORing or adding the ASCII codes) will generate duplicate indices for the two names. In the jargon, there will be a collision.

Collisions are dealt with in two ways, by choosing a function that produces few duplications and, more importantly, by providing a mechanism that will allow the potential duplications to be handled.

There are many functions that can be used to provide an index; I have adopted the simple one of adding ASCII codes quite successfully in a real application (anagrams are not that common). An enhancement to this technique would be to modify each character code, using its position in the string, before adding. This could be achieved by rotating the value a number of times that corresponds to its position in the string, or by multiplying it by that number. The possibilities are endless but space is finite, so I think I'll stop here.

Given an array of a reasonable size, collisions need not be that common, but insurance against them is compulsory.

Insurance

I'm going to supply two basic strategies for dealing with collisions.

The first is to have a secondary data structure behind each location of the primary data table. When there is a collision, each of the entries is placed in this secondary store - this may be a linked list or similar structure - and when a query results in an access to this location, the list is sequentially searched until the correct item is

Windows 3.0 DLL
Now Included!
ROYALTY FREE

Objective dBASE programmers try C++

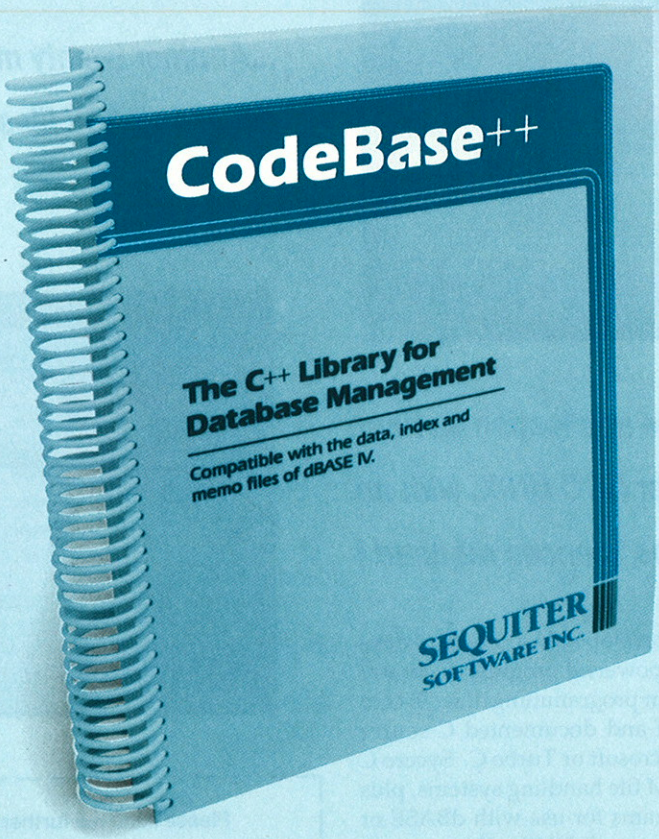
Use CodeBase++ with C++ to build multi-user, dBASE compatible programs. Take advantage of object orientated programming to quickly create maintainable, readable software.

dBASE Compatible

Use CodeBase++ classes to access and change the data, index and memo files of dBASE IV and III. Use other classes to manipulate strings, sort at high speed, perform date arithmetic, evaluate dBASE expressions, manipulate linked lists and manage memory.

Portable

Port your programs between DOS, Microsoft Windows, and OS/2. Keep all the profits as you distribute your programs royalty free.



Fast

Watch as your C++ programs execute much faster than corresponding dBASE, Clipper or FoxPro programs. Be surprised at the small executables and low memory requirements. Examine the C++ source code, which is completely included, to learn good object orientated programming techniques.

Order Today

Order today at £195 by contacting your nearest software dealer. Discover why Sequiter Software Inc. and participating dealers offer a 60-day money back guarantee.

SEQUITER SOFTWARE INC.

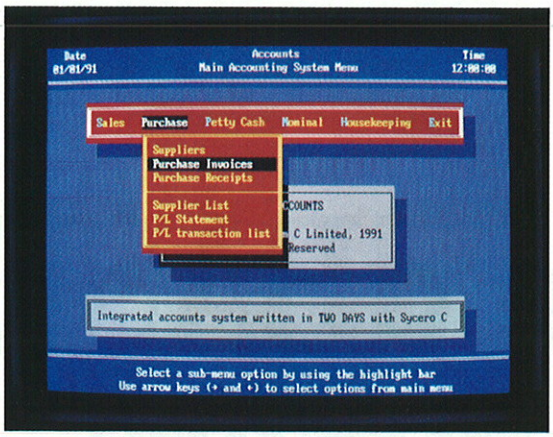
Call 1-403-448-0313
Fax 1-403-448-0315
Hours: 14:30 - 24:00 GMT

P.O. Box 5659, Station L, Edmonton, Alberta, Canada T6C 4G1

CIRCLE NO. 088

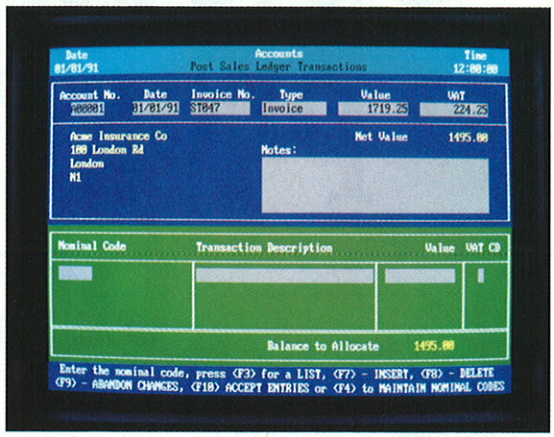
SEE US ON STAND 35 IN EXHIBITION HALL 1 AT SOFTWARE TOOLS '91

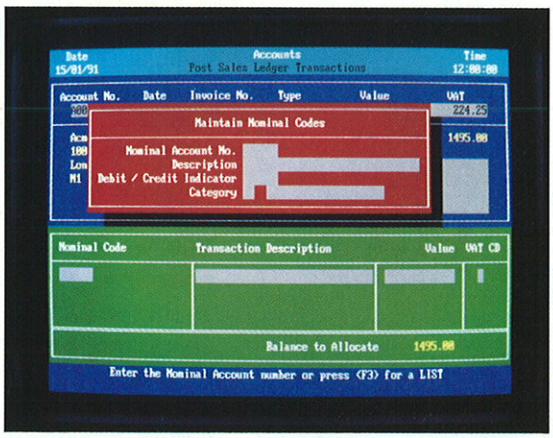
SoftwareDevelopers!



Produce a sophisticated application like this in Microsoft C , with either BTRIEVE or C-ISAM, running under DOS in just a couple of days!

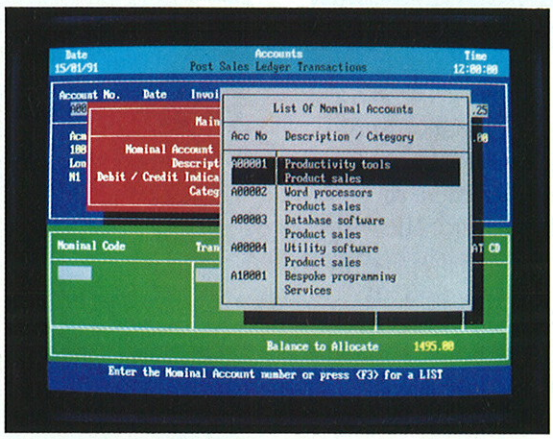
...Here's the same application half an hour later usingTurbo C ++'s advanced VROOM technology...





...Another twenty minutes and it can access dBASE files...

...and here's the same application an hour later running under SCO UNIX, with no runtime licences required !



This entire application was developed in a couple of days using **Sycero C**. Sycero is a powerful program generator that **dramatically** reduces your programming time. Sycero C generates fully structured and documented C source code to be compiled with Microsoft or Turbo C. Sycero C supports the Btrieve/C-ISAM file handling systems, plus it can even generate C programs for use with dBASE or Clipper files and indexes. Sycero C UNIX allows you to develop software to run under SCO UNIX, and you can regenerate existing Sycero C DOS-based applications with absolutely **no changes** required to your definitions.

For more information, either call us on **0622 691616**, or fill out and return the coupon.

CIRCLE NO. 089

dBASE is a trademark of Ashton-Tate. All other trademarks acknowledged.

Please send me further details about Sycero C

Name: .....

Company: .....

Address: .....

.....

.....

.....

.EXE 8/91

Tel: ..... Fax: .....

I am also interested in your generator for Clipper, Sycero dB ☐

System C Ltd 60-61 High Street Maidstone Kent ME14 1SR

Tel 0622 691616 Fax 0622 691241

found. Since the list is likely to be very very short - hopefully no bigger than two - there is no great penalty in implementing this search.

The second method makes use of a circular storage structure. It is very convenient to regard the data table as being circular, this allows the index function to incorporate a modulo *n* division (where the locations in the table are in the range *0* to *n-1*), thus tailoring the final index to the size of the table. When a collision occurs, another integer can be calculated and this used as a 'step' value to move to a new position in the table. If the second position is also found to be occupied then the same step value can be used to go to another position and so on, until a free position is found. In this way a free position will always be found unless the table is completely full. Or will it?

This technique depends on two factors, the first is that the step value must never immediately return to the original position - it must lie between *1* and *n-1* - and the second is that the size of the table must correspond to a primary number.

This second condition is fundamental. Since a primary number is not divisible except by one and itself, any step value will visit all the locations of the table before returning to the starting position. Hence, unless the table is completely full, an empty location is guaranteed to be found no matter how many collisions are encountered.

This assertion may not be immediately obvious but it is true. Think about what would happen if the size of the table was not prime; take a table size of 8 as an example. A step value of two would result in four locations being repeatedly visited and the remainder ignored; a step value of four would result in repeated visits to two locations. It makes no difference from which position you start. Now try a table size of five, any step value under five will visit all locations before returning to the starting point (this can be quickly and conveniently confirmed by counting on your fingers). Note that a step value of one is a special case, this will visit all table locations before returning to the start since it visits them sequentially.

One more point about the step value; it is advisable to use a different method of calculation to that for finding the original index. There are two reasons for this: the step value must be in a slightly different range to the original index (*1* to *n-1*, not *0* to *n-1*) and a different method of calculation decreases the probability that the same

```
/*************************************
 Telephone Directory by Ray Jones 1991

 An illustration of the use and
 implementation of Hash Tables.
*************************************/


#include <stdio.h>

#define OK       1
#define NOT_OK   0

/* Table definitions and declarations */

#define   TABLE_SIZE    11
#define   FIELD_LENGTH  20

typedef struct
{
  char  name[FIELD_LENGTH];
  char  tel_no[FIELD_LENGTH];
} TABLE;


TABLE tel_dir[TABLE_SIZE];

/* end of table defs etc. */

/* Initialise the table with nulls */
void initialise()
{
  int i;
  for(i=0;i<TABLE_SIZE;i++)
    tel_dir[i].name[0] =
       tel_dir[i].tel_no[0] = '\0';
}


/* Hashing function
          - add char values */
int hash(s)
char * s;
{
  char * p;
  int hash_code = 0;
  for(p = s; *p != '\0'; p++)
    hash_code += *p;


  /* Return index value in
                range 0 to TABLE_SIZE-1*/
  return (hash_code%TABLE_SIZE);
}


/* Calculate step value
          - add alternate char values */
int step(s)
char * s;
{
  char * p;
  int toggle = 0,
  step_size = 0;


  for(p = s; *p != '\0'; p++, toggle = ~toggle)
    if(toggle) step_size += *p;


  /* Return index value in
                range 1 to TABLE_SIZE-1*/
  return ((step_size%(TABLE_SIZE-1))+1);
}


/* Find a number from a name */
int dir_query()
{
  int i,h,s, attempts;
  char  name[FIELD_LENGTH],
      num[FIELD_LENGTH];

  printf("\nFind an entry in the directory\n");
  printf("Please enter the name : ");
  scanf("%s",name);

  /* Calculate the hash code and step number */
  h = hash(name);
  s = step(name);

  /* Now search the table for the name */
  for(i = h, attempts = 0;
      strcmp(tel_dir[i].name,name) != 0;
      i = (i += s) % TABLE_SIZE, attempts++)
  {
    /* Check to see if all locations have
                              been checked yet */
    if(attempts == TABLE_SIZE)
      return NOT_OK;
```

Figure 1 - Telephone directory program

step will be generated for symbols that produced the same original index (thus a second collision is less likely).

Telephone Directory

Figure 1 shows a simple telephone directory program using the technique that I have described. The program is written in C and doesn't use any non-standard functions or libraries, so it should run on just about anything.

The `main` function first calls an initialisation routine, (this simply nulls all of the entries in the directory) and then implements a simple menu. The two major functions are `dir_add` and `dir_query`; these implement the basic functions of adding an entry to the directory and looking up a telephone number from a given name. They are very similar in function. The first searches the directory, using the hash and step functions, until a blank entry is found, and the second searches until a match is found between the given name string and that found in the name part of the array. In both cases a check is made to see if the search has covered every location in the array; if it has, then the search is abandoned because either the table is full, or the name being searched for is not in the table.

The hashing function that generates the original index is the one I described earlier; it simply adds each of the character codes in the name string. The step function is similar but adds alternate characters. Both number are treated to a 'modulo *n*' division to bring them into the appropriate ranges.

Within each of the searching functions there are printed messages which show the locations of the array being searched. These would not of course be required in a 'real' system, but they are useful to show the operation of the hashing mechanism. Try entering a few anagrams and see what happens; with any luck, after the first collision the step value will take you to a vacant location. This becomes less likely as the number of entries grows. The array size has been defined to be deliberately small so that you can see this in action without having to enter the entire Yellow Pages or the complete contents of your Little Black Book.

Practically Speaking

The Telephone directory program has been left deliberately simple so as not to hide the technique that I wish to illustrate with unnecessary embellishments. But if you really want to make use of it, simply change the initialisation function to read entries from a file (using the same technique as the `dir_add` function), make sure that the file is saved again on exit if there have been any changes, and change the defined table size to a number that is more useful (not forgetting that this must be prime).

EXE

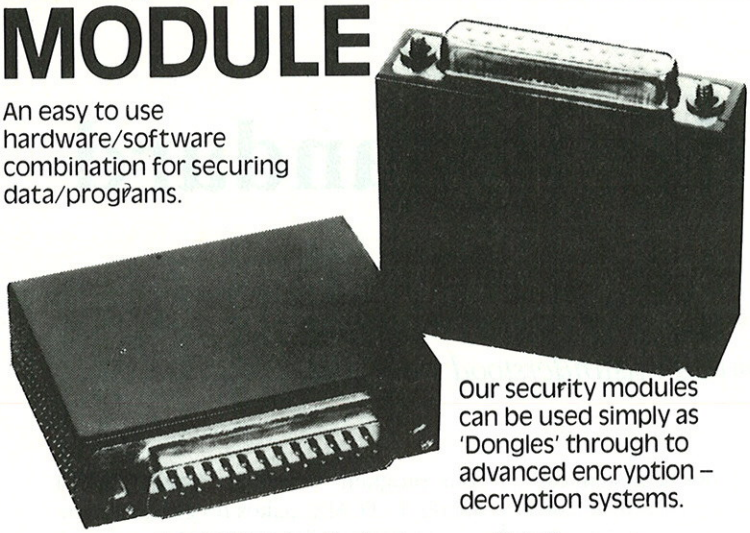
Ray Jones is a software engineer who specialises in the design of technical and real time software. He may be contacted by email at `raj@cel.co.uk`, *or* `raj@uk.co.cel`, *depending on which your mailer likes best.*

```
    printf("Trying location %d\n",i);
  }

  printf("Name found at location %d \n",i);

  /* Found name so print number */
  printf("The number for %s is %s\n",

           name,tel_dir[i].tel_no);

  return OK;
}

/* Add a name and number to the directory */
int dir_add()
{
  int i,h,s, attempts;
  char  new_name[FIELD_LENGTH],
      new_num[FIELD_LENGTH];

  printf("\nAdd an entry to the directory\n");
  printf("Please enter the name : ");
  scanf("%s",new_name);
  printf("and now the number : ");
  scanf("%s",new_num);

  /* Calculate the hash code and step number */
  h = hash(new_name);
  s = step(new_name);

  /* Now search the table for a vacant position */
  for(i = h, attempts = 0;
      tel_dir[i].name[0] != '\0';
      i = (i += s) % TABLE_SIZE, attempts++)
  {
    /* Check to see if all locations have
                          been checked yet */
    if(attempts == TABLE_SIZE)
      return NOT_OK;

  printf("Trying location %d\n",i);
  }

  printf("Free location found %d \n",i);

  /* Found a vacant position so copy in the
                      new name and number */
  strcpy(tel_dir[i].name,new_name);
  strcpy(tel_dir[i].tel_no,new_num);

  return OK;
}

/* Main loop and menu */
void main()
{
char  c[2];

  initialise();


  for(;;)
  {
    printf("\nQuery, Add or Exit? (Q,A,X):");
    scanf("%s",c);
    switch(toupper(c[0]))
    {
      case 'Q':
        if(!dir_query())
          printf("\nName not in directory\n");
        break;
      case 'A':
        if(dir_add())
          printf("\nNew name entered OK\n");
        else
          printf("\nNew name not entered -\
 table full\n");
        break;
      case 'X':
        printf("\nBye for now.\n\n");
        exit(0);
      default :
        printf("\nInvalid selection\n");
        printf("Please press one of [Q,A,X]\n");
    }
  }
}
```

Figure 1 - Telephone directory program (Continued)

SOFTWARE SECURITY MODULE

An easy to use hardware/software combination for securing data/programs.



Our security modules can be used simply as 'Dongles' through to advanced encryption – decryption systems.

- ★ For IBM PC family (and compatibles)
- ★ Variants for other systems
- ★ Advanced construction – rugged and compact
- ★ Proven throughout the world
- ★ Transparent operation – user friendly
- ★ Driver software supplied for variety of applications

Control Telemetry of London
11 Canfield Place, London NW6 3BT
Tel: 071-328 1155 Fax: 071-328 9149

CIRCLE NO. 090

PRO-DIR V3.0

The Power Directory Utility for the Professional Programmer

NO MENUS !!!
45 page manual and on-line help included

PRO-DIR brings you the power of a command line directory utility with unsurpassed search and selection features. It combines some of the best features seen on DOS, UNIX, VMS etc. at an affordable price of just £49 (including VAT and P&P).

PRO-DIR makes the task of finding and managing files effortless, even on large directory structures. You can now find those long lost files on your network drive with commands like

```
d  G:\USER\BOB\...\TAX*R*T*N.*      after=01-01-87      before=-365
```

Look how easy it is to recover disk space when your hard disk is full and you need to free a few megabytes quickly - the command

```
d  C:\,D:\,E:\      +R      min=50k      since=-7
```

finds all files larger than 50kb that you created over the last seven days *anywhere* on drives C, D or E. (Also see below how the +B switch can help).

TAKE A LOOK AT THESE FEATURES

```
d  *A*B*02.E?E          Extended wildcards in all file and directory specs
d  *.EXE,*.BAT,*.COM    Multiple unlimited file select specs
d  *.*      *.EXE,*.BAT Multiple unlimited file exclude specs
d  C:\*\*S*R*C*\*.C     Extended wildcard directory specification
d  C:\*.C,D:\USER\*.C   Searching on multiple drives (netwo
d  C:\USER\*.C    +R    Recursively find all matching
d  \A*\*.C              Search for files that ap
d  \A*\...\*S*R*C*\...\*.C  Search for files
d  +S:f                 Sort by file
d  max=2M  min=50k      Selec
d  on=-7
d  after=-365  before=-3
d  after=10-12-90
d  a=h  n=a
d  +P
d  +B
d  +T
d  +h
d  +m
d  +ft-as
d  +?
set PDOPT
```

directories

ories

date.

uden file)

ar processing

files

the selected files

may fields. (eg f=filename field, s=file size field etc.)

line help screens

customization of program default settings via the set command

JUST £2
BRINGS YOU A DEMO COPY
The Demo Copy is fully functional, except that file information after the 10th file is replaced by asterisks. Please specify disk size with orders.

You can com...ures together. You may specify as complicated a command as you wish, provided it fits on the DOS c...e! And that's not all - we haven't even listed all the switches and options that are available. Call or fax us f...more information. Alternatively, send us the coupon below, or call us with your order 9AM - 5.30PM.

Please send me ..... copies of PRO-DIR at £49 per copy including VAT and P&P.
Name: .....
Address: .....
.....Phone: .....
Access/Visa number: .....
Signature: ..... Expiry date: .....
Please specify disk format: ___ 5.25" (360kb) OR ___ 3.5" (720kb)
If credit cardholder name and address are different from above, please specify. If paying by cheque, please make it payable to Korala Software. Thank you.

KORALA SOFTWARE
Korala Associates Ltd
FREEPOST
Edinburgh EH3 0EP
Phone: 031 556 9208
Fax: 031 556 9215

Suitable for IBM PC, XT, AT or 100% compatibles running DOS 3.0 or higher. Please allow 28 days for delivery. International orders: No VAT, postage charged at cost.

CIRCLE NO. 091

System Science

C COMPILERS

Microsoft C 6.0 with PWB	£225.00
Quick C & Quick Asm 2.51	£95.00
Quick C ver 2.5	£65.00
Borland C++	£229.00
Zortech C++ Dev Ed 3.0	£400.00
Watcom C 8.0	£255.00
Aztec C86 Comm	£265.00
C compilers for 386	£call

C GRAPHICS & DATAFILE

Halo	£195.00
GSS GDT	£475.00
Metawindows	£195.00
Essential Graphics (no roy)	£245.00
PCX Prog Toolkit (new 4.0)	£165.00

C DATA & MISC

Btrieve for DOS	£345.00
XQL for Btrieve	£475.00
Ctree Plus	£345.00
CodeBase 4.2 (source)	£175.00
CodeBase ++	£175.00
C Lisp	£245.00
Blaise C Tools Plus	£89.00

C SCREENS & COMMS

Cscape, Look & Feel	£365.00
Vermont Views & Designer	£325.00
Panel Plus II	£275.00
C Worthy & cwARCHITECT	£call
Essential Comms (src)	£195.00
Blaise C Aynch Manager	£115.00
Greenleaf CommLib	£185.00

PROGRAMMERS TOOLS

SPE prog editor	£175.00
MKS Toolkit (Unix toolset)	£155.00
PVCS Prof	£345.00
PolyMake	£129.00
CProgrammer Toolbox	£159.00
Brief Editor 3.0	£195.00
Kedit Editor	£110.00
Flow Charting 3	£150.00
Dan Bricklin DEMO II (new)	£195.00
MS Progs Lib 1.2 CD	£225.00
Personal Rexx	£110.00

ASSEMBLERS & LINKERS

MS Macro ASM 6	£75.00
Turbo Debugger & ASM 2.0	£99.00
RTLink Plus	£345.00
Plink 86 Plus	£295.00
Sourcer & Bios (DisAsm)	£125.00

CROSS DEVELOPMENT

Introl C or Modula-2	£1445.00
Aztec C Cross	£275.00
IAR C Cross with CSpy	£call
2500AD Cross AsM	£135.00
Simulators & others	£call

FORTRAN

MS Fortran-77 5.0	£235.00
Lahey F77L-EM/32 & 386 OS	£845.00
Salford FTN77/386	£765.00
Lahey F77L	£375.00
RM Fortran Forte	£475.00

PASCAL, BASIC

Turbo Pascal (6.0)	£75.00
Object Professional (for TP)	£110.00
Turbo Pascal Windows	£115.00
Blaise Power Tools Plus	£89.00
Blaise Asynch Plus	£89.00
MS Basic 7.1 with PWB	£225.00
MS Quick Basic 4.5	£60.00
PowerBasic (ex Turbo Basic)	£85.00
MS Visual Basic (New)	£99.00
many Pascal, Basic libraries	£call

OTHER LANGUAGES

Smalltalk V/Windows	£295.00
TopSpeed Modula-2	£115.00
MS Cobol 4.0 (new ver)	£399.00
RM Cobol-85 ver 5.0	£995.00
LMI ur Forth	£295.00
PC Logo	£50.00
Clipper 5.0	£395.00

MEMORY, MATHS etc

386 Max 5.0	£75.00
QEMM/386 5.1	£55.00
QRAM for 286	£45.00
Above Disc (LIM Emul)	£69.00
Norton Utilities (new ver 5)	£110.00
Spin Rite II	£62.00
Derive (Symbol Alg) 2.0	£135.00
MicroStat II	£295.00
Mathematica by Wolfram	£call

WINDOWS 3.0

MS Windows 3.0	£85.00
MS Windows SDK & MS-C	£425.00
MS Windows SDK 3.0	£295.00
Case: W New Corp User	£CALL
WinPro/3 from Xian	£525.00
WindowsMaker	£595.00
Toolbook by Asymetrix	£295.00

"OUR RANGE INCLUDES MANY MORE ITEMS FOR DOS, OS/2, UNIX ETC. PLEASE ENQUIRE"

3-5 Cynthia Street,
London N1 9LF
Fax: 071-837 6411

Prices are subject to change and alteration Prices include UK Delivery but are exclusive of VAT

071-833 1022

CIRCLE NO. 092

POSIX - consider it standard

What is POSIX? What is it for? How does it differ from UNIX? Here is Peter Collinson on a much-misunderstood standard.

Someone from a local council rang me up the other day saying 'I've read your articles in *.EXE* and wondered where I could get copies of the POSIX standards document from?' This was interesting because in the last few months I have become involved with the IEEE efforts in standardising UNIX and creating the POSIX standards. Some of the time I am working for the US UNIX User group, Usenix, as their Standard's Liaison. I guess that the magic word 'POSIX' is beginning to appear on system purchase specifications.

The Institute of Electrical and Electronics Engineers (IEEE) is a US group that has been involved with the generation of many standards. The standardisation work that sprang from UNIX is more usually known as POSIX, the Portable Operating System. There are a couple of common misconceptions. First, the standards do not define UNIX, they define POSIX. One way to think of this is that POSIX is a generalisation of UNIX. Many UNIX systems are POSIX compliant and it's quite likely that other systems with different names will also measure up to the standard.

Second, it's not a single standard. It's a group of about twenty standards; or it will be, because only two have reached the end of the process. It's a group of standards because of history. When the IEEE established the controlling committee for POSIX in 1985, the work was labelled project 1003, or P1003. This quickly split into two, P1003.1 for the operating system interface and P1003.2 for the shells and utilities. It is current practice to replace the 'P1003' by 'POSIX', so the original projects are known as POSIX.1 and POSIX.2. There are now around 20 active projects, and knowing which 'dot' number relates to which activity has become an expert subject suitable for Mastermind questions.

What most people think of as 'POSIX' is actually 'POSIX.1'. This defines a set of routines that may or may not be system calls. They specify the system interface, the interface that applications use to make things happen in the outside world. POSIX.1 is now an international standard, printed on A4 paper rather than quarto, with a new grand international number. The standard is: ISO/IEC 9945-1: 1990, Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C language]. The 'C language' in brackets means that the standard is expressed in a specific language - yes, C. ISO would much prefer this to be independent of any language and work is under way to do just that.

The second project, POSIX.2 (shells and utilities) has not managed to generate a full standard yet, but it is close. A third project, POSIX.3, produced a standard in April of this year. It's called 'Test Methods for measuring Conformance with POSIX' and 'defines general rules for developing test assertions and related test methods for measuring conformance of an implementation to POSIX standards'. The standard is really aimed at the people who make test suites and the people who write standards for the testers to test. All the other projects are at different stages of development. Perhaps I will do an article on the proliferation of POSIX in a future column.

What does all this mean?

We should be seeing systems popping up that claim to be POSIX.1 compliant. Strictly they are not unless they have been tested using a verification suite that complies with POSIX.3, but this will take a little time to filter in.

As (and if) POSIX systems become more widespread it should mean that programs written on one system should transfer trivially to another. To ensure that this can happen you will have to make sure that your program is written to the POSIX.1 standard - POSIX makes no guarantee unless you follow all the rules. The goal in all this has never been binary portability, but the ability to move source from one system to another, compile it and go.

One effect that I am hoping to see is the widespread adoption of the ANSI C standard. Standards-makers like to base their standards on other definitions, and POSIX is no exception. It has taken the ANSI C language definition and routine set as a basis for moving forward. If a system is POSIX compliant, then it should offer ANSI C.

It has been annoying to see the push to sell new things arm the DOS world with ANSI C, when many UNIX systems are still supplied with C compilers based on the old `pcc` compiler (or variants of it). I want the common base of everyone to be ANSI C, so that this eliminates any fear of non-portability when you write code for someone else. To achieve this, everyone buying a new UNIX system must be *given* an ANSI C compiler.

Basically, I want to write in ANSI C without all the disgusting ruses needed to make the C-preprocessor remove the function specs so that things will compile on old C compilers. If I am to write in ANSI C, I want it to be widespread so that everybody who has a C compiler can compile my programs. I am hoping that POSIX has this effect at least.

Changes to the system

POSIX.1 started life as a codification of UNIX System III and then System V. It also has had significant input from the Berkeley world. POSIX.1 does genuinely attempt to join both the System V and BSD worlds and is stronger by taking the experience from both camps. This makes things a little hard for System V, because it has always guaranteed backwards compatibility. A number of

Xvale Ltd., Store 'D'
55, Bedford Court Mansions,
Bedford Avenue, London.
WC1B 3AD U.K.

TEL: 071-636-8210 / 071-631-4818
FAX: 071-255-1038 24 Hours

Barclaycard
Access

UNIX

Now! you can get into UNIX and VPIX (Multi-user DOS) at affordable prices, with a leading specialist of the UK.

Interactive Unix SysV, REL 3.2
Architect Series: (Ver 2.02-2.21)

SysV,Rel 4
CALL !

Workstation Developer: 1-2 usr £1295.00 U/L usr £1995.00
Operating System, Vpix (DOS under UNIX),
Software developement system plus New 'C', CO-edit, Codewatch,
Ten-Plus, Looking glass, XII windows runtime and developement system,
NFS/TCP/IP, and complete documentation.

Workstation Platform: 1-2 usr £855.00 U/L usr £1495.00
As workstation developer, excluding networking and developement.

Network Developer: 1-2 usr£ £1155.00 U/L usr £1815.00
As workstation developer, excluding XII items.

Network Platform: 1-2 usr £695.00 U/L usr £1350.00
As Network Developer, excluding developement items.

Application Developer: 1-2 usr £1095.00 U/L usr £1750.00
As Network Developer excluding network items.

Application Platform: 1-2 usr £495.00 U/L usr £1095.00
As Application Developer, excluding developement items.

UNIX ITEMS	
Motif Dev	£345.00
PC-DOS bridge	£195.00
Norton for unix	£195.00
Easy windows	£595.00
Ported Netware	£3995.00
SMB/ix	£895.00
Informix/Oracle	Call
Lotus 123	£595.00
LPI Cobol	£1095.00
LPI New 'C'	£595.00
LPI Fortran	£ 895.00
LPI Pascal	£895.00
LPI Basic	£595.00
LPI /PL/1	£1795.00
LPI RPGII Dev	£1795.00
LPI RPGII Util	£895.00
MENUIX	£49.95
SYSIX -pro	£145.00
LPI (unix) C++	CALL
XSIGHT	CALL

OTHER ITEMS	
Xenix Items	Call
Unix o/sys	£525.00
Unix s/dev	£613.00
Unix vpix	£321.00
Unix tcp/ip	£227.00
Unix nfs	£340.00
Open dsktp	£795.00
O/dtp s/dev	£1095.00
O/dtp servr	£1095.00
Guru reprter	£49.95
Syseditor	£49.95
Cshell	£49.95
Menuix	£49.95
Sysix Pro	£145.00
SystemIX	£295.00
PC-Connect	£450.00
Pc-interface	£170.00
PC-TCP	£375.00
PC-Xview	£245.00
Terminl ctl	£695.00
Xvision	£295.00

LNX

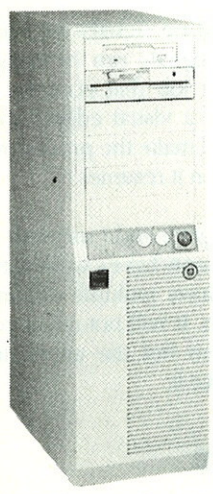
*Two hardware configurations + Software (**a-e :FREE!.**)*
*1) Entry level minimum mono hardware systems (**a-b: free**)*
*2) Unix level higher spec hardware systems (**c-e: free**)*

Free Software Bundles

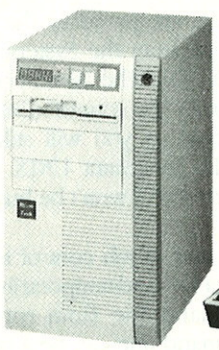
a) Dos 4.01,Menuix,C-shell,Sysed under DOS, £225
b) Unix,Sysix,Sysed,Dos, £475
c)+b)Appl-Platfrm v2.02, . £770
d)+b)Net/Wrk-Plat v2.02, £1130
e)+b)Wrkstn-Dev v2.02 £1570

HD/M'board	386-20	386-25	386-33	486-25	NB386-16	NB386-20
40 Mb	£895a / £1445b	£945a / £1695b	£1195a /£1845b	£1995b/£2790c	£1695b	£1895b
80 Mb	£1040b/£1690b	£1085b/£1810b	£1340b/£1995c	£2140b/£2935e	£1995b (60mb)	£2195c (60m)
120 Mb	£1110b/£1760c	£1225b/£1975c	£1410b/£2060d	£2210b/£3075e	** *PRICE INFO* ** Entry level followed by Unix level,. (a-e) = free software	
200 Mb	£1185b/£1835d	£1300b/£2050d	£1475b /£2325e	£2275b/3200e		

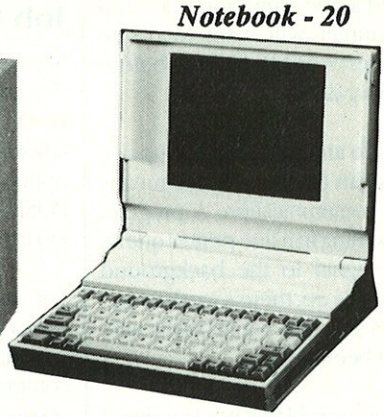
Unix level systems: Mini-tower,1024*768 SVGA Colour screen,102 keyboard,SVGA card w:512k ram, 2Ser/1Par, Floppy/hard ctrlr, 5-1/4 AND 3-1/2 drives, 4MB RAM(486=8mb), Ami-Phoenix-Award BIOS, 12 mth RTB wty.



LNX systems



Notebook - 20



Entry level systems As for Unix except.: 1 floppy 1 Mb ram Mono screen & adapter

decisions in the system design have been coloured by that need.

The first change to the System V view of the world is in the area of signals. A signal is an event sent to a process to indicate that something unexpected has happened. This is perhaps floating point overflow from the hardware; or the user has typed the interrupt key and wants to stop the current process running. The interrupt key is often Control-C, but it can be changed by the user. UNIX gives you a well defined mechanism for handling these events.

A process running on UNIX chooses what to do when signals are received. First, it can ignore their existence and leave them set to their default values. The default action that is taken depends on the nature of the signal; mostly the process will die. Most programs do this. Second, the process can take action to ignore the signal completely.

Finally, the process can arrange to catch the signal. This means that a routine is nominated to be called when the signal occurs. The UNIX program will say:

```
int func();
....
signal(SIGINT, func);
```

This makes the program call the `func()` routine when the SIGINT signal is received. SIGINT is the signal that is sent when the user hits the interrupt key.

The program merrily pursues its course until the user strikes the interrupt key. The system arranges that the routine `func()` is entered and does the stuff that the user wants. Exit from `func()` will make the process resume execution from where it was rudely interrupted.

However, most catching routines do some work and then exit from the process. The signal has been caught to tidy up the world a little before exiting. Commonly, the process has created some temporary files and a good programmer will want to delete them, being a responsible citizen, before the process finally dies.

Signals started life aimed at the tidy-up application but began to be used as a means of interprocess communication. A program that placed something in a line printer queue would send a signal to the background process to say 'wake up, there's work to do'. It was then that a snag was found in the way that signals had been defined.

The original systems said that when a signal had been taken, the action of the signal reverted to the default. The catcher routine would be called, and would need to reset the action of the signal back to program control. Signal catching routines contained code like:

```
func()
{
    signal(SIGINT, func);
    ....
```

to reset the signal handler when the signal occurred, or perhaps

```
func()
{
    signal(SIGINT, SIG_IGN);
    ....
```

to ignore the signal until later.

The problem is that there is a finite time *after* the first signal was received before the `signal` routine can be called to reset things. In the interim, the action of the signal is set to the default: sudden death. So a second signal can arrive quickly after the first and the process is killed because it has not had sufficient time to reset the signal handler.

POSIX.1 has adopted a solution from 4.3BSD that allows signals to be treated very much like traditional interrupts. Signal handlers are permanently installed. Signals can be blocked from delivery by use of a *signal* mask. A process also has a set of pending signals. These include any signals that might have been generated but are currently blocked from delivery. If a signal occurs that is blocked, it is added to the set of currently pending signals.

A process can set the mask at any time and can also specify what the mask should be set to when a signal happens. The signal handler can be entered with the default masking state of the signals already established.

The result of all this is to make signals considerably more predictable and it is now possible to write event driven programs where events are triggered by signals.

Job Control

Signals are probably the area of greatest change from the System V view of what UNIX should be. Another is the implementation of 'job control'. Unfortunately this is optional so you will still be able to get POSIX compliant UNIX systems without job control. I won't be buying them.

The idea of job control is that a user can type a key and temporarily stop the current command line from running. Since the command line can be a collection of processes, the word 'job' is used to mean 'the set of processes that constitute a single task'. Of course, this is most often just a single process. To get a flavour of job control in action, let's think what happens when you are compiling a huge C program:

```
cc -O -o huge huge.c
```

and the boss walks in to ask a question. The question means that you need to look at something on the computer. If you don't have windows or separate pseudo terminals then you are in a quandary. Do you kill the compilation to answer the question? Has is it nearly finished? Can you keep the boss talking long enough? All these thoughts rush through your brain as you reach for the interrupt key to kill the compilation.

With job control you simply type Control-Z to suspend the job and then use the `bg` command to continue the job running in the background. With `csh` it looks like:

```
% cc -O -o huge huge.c
^Z
Stopped
% bg
[1] cc -O -o huge huge.c &
%
```

The sequence allows you to move a command from the foreground to the background so it can continue executing leaving the foreground free for other things. In this case answering that stupid question (calm, calm). When the boss leaves, you can move the job into the foreground by saying

```
% fg
cc -O -o huge huge.c
```

the shell will reprint the command and move it into the foreground.

There are several reasons why this approach works well. The first is that I can suspend the current job without planning ahead. I don't have to worry about creating a new window. I don't have to have already created a pseudo-terminal or shell layer. I just simply type Control-Z and get on with things.

Secondly, it's based on signals. As I have described above, signals can be caught. When I suspend a job that controls the whole screen (like a visual editor) I can catch the signal and make the process redraw the screen when it resumes.

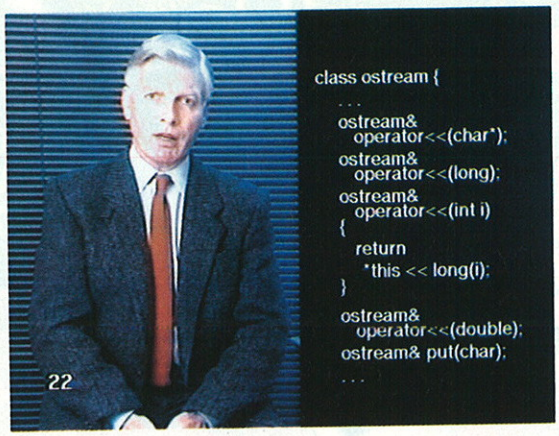
This system is just great and I use it all the time, even though I now have a workstation and the ability to have multiple terminals on the same screen. It will not affect your application program but the application programmer will like it.

More reading

I have picked on perhaps the two most interesting areas to highlight. I should also warn BSD programmers that they are going to need to alter their ideas of how to control

VIDEO COURSE

Learn C++



NO RUSH!
NO TRAVEL!
NO HOTELS!
ONLY £299.95

Learn C++ Now!
The great exodus of programmers from C to C++ has begun! Since C++ builds on C, it's the easiest OOP language to learn. That's why it's called "the language of the 90's".

Why the rush?
PRODUCTIVITY! Yes, C++ programmers can write programs in less time requiring less maintenance. Large projects become much easier to manage.

Unfortunately, learning C++ can be very costly. Classroom instruction is expensive even without the travel and hotel costs. Of course, not learning will cost you even more in the long run. Now there is an alternative!

The top C++ video tutorial at the lowest possible price.
The C++ video tutorial from Zortech is the ultimate C++ training tool for work or home at only £299.95. It comes on six VHS video tapes containing 32 clear, extensive tutorials.

Used in conjunction with the concise workbook and tutorial disk, you will find everything you need for fast-track C++ tuition.

The course is generic (i.e. compiler and hardware independent) and is available with or without the award winning Zortech C++ Compiler for MS-DOS and OS/2.

Start writing C++ code within a week.
As a C programmer, you will start producing C++ code within a week of concentrated use of this course. Alternatively, spend just an hour a day watching the video and working through the suggested exercises to learn C++ in only six weeks!

The Leader in video tutorials.
If you don't already know C, you can join tens of thousands of programmers world-wide who have learnt C with the Zortech Complete C Video Course, described as:

"An excellent bargain ...
... I heartily recommend"
Gary Ray, PC WEEK

Save your company thousands of £££'s.
One programmer can train for only £299.95, but you can train ten programmers for just:

C++ Course	£299.95
9 Extra Workbooks	£179.55
Total 10 students	£479.50

Yes! Only £47.95 each! (With all the FREE refresher courses you need!)

Now, Zortech with its new C++ Video Tutorial has refined the art of video tuition and presentation even further for the 1990's.

Only £299.95 complete
- Six Videos with 32 lessons
- 256 page workbook
- Tutorial disk
- Compiler & hardware independent
- NTSC or PAL format
- Tax deductible

Don't delay, order now!
Just mail the coupon or call the order hotline for same day shipment.

USA: Zortech Inc.,
1165 Massachusetts Avenue,
ARLINGTON, MA02174
Voice: 617-646-6703
Fax: 617-643-7969

EUROPE: Zortech Ltd.,
106-108 Powis Street, LONDON
SE18 6LU
Voice: 44- 1-316-7777
Fax: 44- 1-316-4138

IN EUROPE CALL:
44- 1-316-7777

ORDER FORM
Please rush me these items:

Qty	Description	Price
___	C++ Video Tutorial	£299.95
___	Extra C++ Workbooks	£. 19.95
___	Zortech C++ Compiler	£129.95
___	Developer's Edition	£299.95

Please add £5 + VAT postage in UK
All UK orders please add 15% VAT
International shipping charged at cost

Name _____
Company _____
Address _____
_____
_____
Phone _____
MC,VISA or CHQ_____ Expiry ____
Card No. _____ .EXE 7/91

CIRCLE NO. 094


DOS POWER TOOLS
PAUL SOMERSON
TECHNIQUES, TRICKS AND UTILITIES
REVISED FOR DOS 5.0
2ND EDITION
3 DISKS INCLUDED
5.0
OVER 100 BRAND NEW UTILITIES ON THREE DISKS
The all-time worldwide bestseller, now revised and updated for DOS 5.0
'A fabulous work, in a league by itself... a must buy' John Dvorak
THE BEST JUST GOT BETTER.
BANTAM COMPUTER BOOKS
THE POWERHOUSE BEHIND POWER TOOLS
1-2-3 POWER TOOLS
WINDOWS 3 POWER TOOLS
NOVELL NETWARE POWER TOOLS
MIKE EDELHART
All available now from your local computer bookseller, or order direct on our 24-hour credit card hotline 0326 77781
For further information, please contact Stephen Page, Sales and Marketing Manager, Bantam Computer Books, Transworld Publishers Ltd, 61-63 Uxbridge Road, London W5 5SA
BANTAM COMPUTER BOOKS
Bantam Computer Books is a Division of Transworld Publishers Ltd.
CIRCLE NO. 095

terminals. The System V terminal interface is now standard (there are some slight changes). It's very hard in a small article to cover what the changes will mean to a programmer. If you feel that POSIX will affect you then you can find some further reading.

The standard itself, 'ISO/IEC 9945-1: 1990, Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C language]' is published by the IEEE and available from them in the US. It is available in the UK from ILI (London Information) phone 0344 23377. You must ask for P1003-1990 and you will get a price of £58. They will charge £103 if you ask for ISO/IEC 9945-1 (strange). The document *is* expensive. It's also a little impenetrable since it's designed to specify something rather than teach you how to use it. It does have two sections, the first is the standard itself and the second the rationale behind many of the features. These are numbered together so if you want to look up the rationale behind section 7.2.4 you look this up in B7.2.4.

Then there is a book in the Nutshell series from O'Reilly & Associates. The book is 'POSIX Programmer's Guide' by Donald Lewine. Its ISBN is 0-937175-73-0. The book is distributed in the UK by Addison-Wesley.

The book is split into two sections, the first third contains ten chapters of discussion. This is followed by 300 pages containing the description of the library functions that a programmer will use. The second part of the book is done well, and will be very useful as a reference guide. I especially liked the addition of references to the standards on each 'manual' page. There are several appendices and a comprehensive index.

I think that the first section of the book is adequate but very thin in many places. For example, the chapter on processes has a lot of discussion on the system calls that exist and only a small amount giving the why and the how. The book avoids much mention of job control, this is relegated to a couple of pages at the end of the chapter on terminals. The book really fails when things get more 'technical'. However, it is well written and is easy to read.

The final book is along the same lines as the last. This is 'The POSIX.1 Standard, A Programmer's guide' by Fred Zlotnick. This is published by Benjamin/Cummings, ISBN 0-8053-9605-5. and is also distributed in the UK by Addison-Wesley.

This book is much more complete from a technical point of view. It gives a lot more background and explanation of why decisions were taken, it does more comparison with existing reference systems (System V.3 and 4.3BSD). I like the way that every routine specification is shown with any associated header files.

I much prefer Zlotnick's book - it seems more complete and contains more information. It *is* much denser, and possibly harder to read. However, the reference material is not as good as Lewine's book and I would guess that if I started programming to the standard then Lewine's book would lie about the desk being used as reference material. I am pleased to have both books.

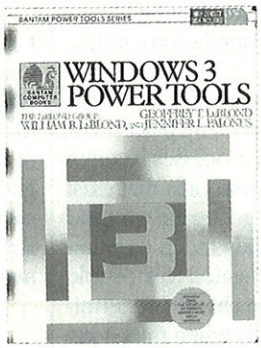
EXE

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

Books

Power Users and Powerful Processors.

Windows for Who?



Windows 3 Power Tools. Crikey, I never thought I'd need a GUI to use my Black and Decker. This tome is dedicated to a breed of Windows junkies known as 'Power Users'. These people apparently just can't keep their hands off the thing and are thirsting for every last drop of info about how to muck about with it. It is not designed for Windows programmers but rather for experienced Windows (and computer) users. There are 17 chapters in the book, the first 13 dealing with tips and techniques for fine-tuning the operating environment. The last four chapters are concerned with the software that comes on a diskette with the book.

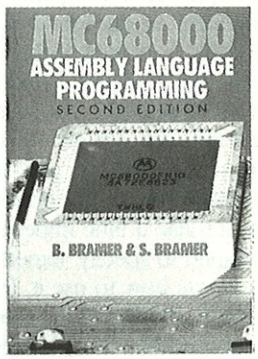
The main text of the book can be divided into two categories: statements of the glaringly obvious; and the bits that might be useful. For instance, the chapters on Program Manager and File Manager Techniques don't appear to tell you much you couldn't glean from dipping into the Windows manual. The chapters on printing, configuration, and sharing data between applications may contain some tips that these power users are not aware of, but I don't know. The problem is, you see, that I'm not quite sure what the people this book is written for actually get up to. One gets the impression that they must be closet boffins masquerading as businessmen. The level of expertise assumed in the reader vacillates wildly. Topics covered range from discussions about Windows memory management and extended memory drivers to how you arrange icons in the program manager. Consequently the snippets of handy information are embedded in a lot of turgid waffle.

The disk that is supplied with the book contains four applications that allow you to further customise (ie tamper with) your generic copy of Windows 3. These are: *Oriel*, a graphics-based batch language that lets you build your own Windows apps; *Command Post*, a character-based replacement for the File Manager (what's the point ?); *Aporia*, an 'object-oriented' (I use the term loosely) graphical shell for Windows; and *IconDraw*, an icon drawing program that lets you design your own custom icons for applications (whoopee). As far as I can make out, none of these are of any use and would, if installed, utterly confuse anybody else who jumped on your machine. Surely, one of the main benefits of using something like Windows is that the shell presents a consistent system view to any user. I have no doubt that the programs are well written and very clever but I do question their *raison d'être.*

Windows 3 Power Users, come out, come out, wherever you are! This book may be the one for you, but somehow I think not.

Title: Windows 3 Power Tools *Price: £46.99*
Authors: Geoffrey T. LeBlond, William B. LeBlond and Jennifer L. Palonus *Pages: 664*
Publisher: Bantam Computer Books *ISBN: 0-553-35298-9*

A Tale of One Processor



Has the segmented architecture finally taken a toll on your sanity? Motorola has been producing the 68000 since 1979 and offers a genuine alternative to Intel's 80x86 family of microprocessors. Speaking as a born and bred MS-DOS programmer, reviewing a book on the 68000 seemed like the most appropriate way to acquire a genuine insight into the fundamentals behind this processor.

MC68000 Assembly Language Programming is intended as an introduction to assembly language programming for first year Computer Science students and assumes the knowledge of a high-level language like Pascal or Modula-2. Each chapter may be treated independently and aims to give the reader a thorough understanding of the 68000 repertoire, ending with a set of exercises; luckily for the discerning student, Appendix G contains a set of answers. On closer examination, it would seem that some of the questions given tend to be rather monotonous. Don't be surprised if you are asked to rewrite a previous exercise using a different addressing mode or using a new instruction. Adding a little variety would have certainly been more entertaining.

There are frequent reviews of what has been introduced so far and this provides a painless method of determining which subsection contains information on a particular subject. With the evidence given so far, you may be under the impression that *MC68000 Assembly Language Programming* is purely a tutorial guide for delinquent Computer Science students, but there is more. The complete 68000 instruction set is covered in detail, together with numerous example programs. There are even chapters devoted to programming the main peripheral devices in the Motorola family. These are presented in the format of a technical data sheet and contain example code. Both polling and interrupt driven I/O are given adequate coverage.

As a passing observation, you should be warned that the examples in these chapters tend to be annoyingly similar. For instance, you are not only subjected to serial I/O on two functionally similar devices (the MC6850 ACIA and the MC68681 DUART) but identical examples using both polled and interrupt driven serial I/O are also given. Most of the example programs are written for a number of single board computers but I am sure that it is not too difficult to adapt the code to run on your STs, Amigas or Macs. For the owners of such machines *MC68000 Assembly Language Programming* provides an adequate reference covering the programming of the 68000 microprocessor and its peripherals, whereas anyone wishing to learn 68000 assembler would do well to consider this offering in terms of its tutorial benefits.

Title: MC68000 Assembly Language Programming *Price: £16.95*
Authors: Brian Bramer & Susan Bramer *Pages: 344*
Publisher: Edward Arnold *ISBN: 0-340-544-51-1*

Books Received This Month

Object Oriented Assembly Language by Len Dorfman	Windcrest	£18.95	ISBN:0-8306-7620-1	pp360
A New Guide To Artificial Intelligence by Derek Partridge	Albex	£19.95	ISBN:0-89391-608-0	pp546
Novell Netware PowerTools by R J Palonus	Bantam	£46.99	ISBN:0-553-35298-9	pp664
LOGIC - A Foundation For Computer Science by V Sperschneider and G Antoniou	Addison Wesley	£21.95	ISBN:0-201-56514-5	pp495

"With or without a mouse, it's the cat's whiskers."

New development environments and technology transition make 80's editors inefficient and obsolete. To be competitive, programmers must have leading edge tools. The tool you'll use the most, the one that wraps everything into a single consistent environment will be the powerful, configurable SPE Professional Editor.

Right out of the box, it'll make you more productive, more adaptable, more imaginative. Its instant installation, elegant mouse support, advanced user interface, and point-and-shoot help get you running immediately.

MS-DOS, OS/2 and Dual Mode together

MS-DOS, OS/2 and Dual Mode versions are all included in the same box on both $5\frac{1}{4}$" and $3\frac{1}{2}$" disks, with templates for popular languages, with or without a bundled Microsoft® Mouse. Great value, and your team can move into the 90's with the same editor.

Turnkey emulations

If you prefer the commands and keystrokes of other editors, our turnkey emulations duplicate them precisely. And you still gain the SPE Professional Editor's advanced features, windowing capabilities and powerful engine.

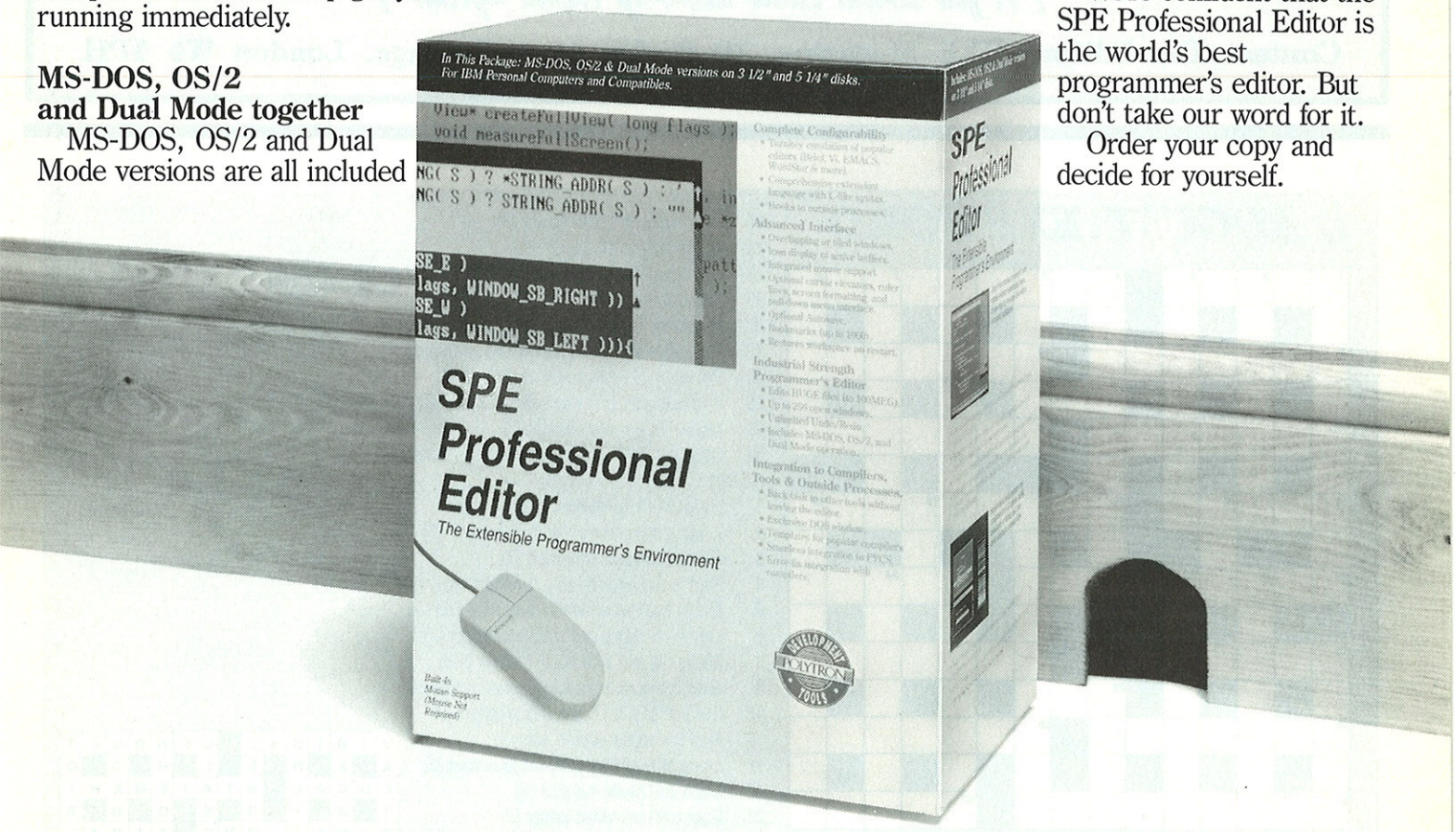
With or without a mouse

Use the editor with or without a mouse — all functions are available without lifting your fingers from the keyboard. But the click-select windows, scroll bars, zoom, shrink-to-icon, block text selects, and other speedy mouse actions make editing extremely efficient.

Don't take our word for it

We're confident that the SPE Professional Editor is the world's best programmer's editor. But don't take our word for it.

Order your copy and decide for yourself.



Works with or without a mouse.
Packaged with or without Microsoft® Mouse.

Version 1.1 now available

APPROVED DEALERS

Absolute Computing Limited, Absolute House
124 Leavesden Road, Watford, Herts. WD2 5EG. Tel: (0923) 242570

Computer Solutions Limited, Canada Road, Byfleet
Surrey KT14 7HQ. Tel: (09323) 52744

Grey Matter Limited, Prigg Meadow, Ashburton
Devon TQ13 7DF. Tel: (0364) 53499

Roundhill Computer Systems Limited, Orchard House
Ogbourne St. George, Marlborough, Wiltshire SN8 1SU. Tel: (0672) 84535

Reflex Technology Limited, 9 Buckingham Place
Bellfield Road, High Wycombe, Bucks HP13 5HW. Tel (0494) 465907

Frontline Distribution Limited, Intec 1, Wade Road
Basingstoke, Hampshire RG24 0NE. Tel: (0256) 463344

System Science, 3-5 Cynthia Street
London N1 9JF. Tel: 071-833 1022

Software Express, Portmill House, Portmill Lane
Hitchin, Herts SG5 1DJ. Tel: (0462) 422525

Software Generation · Portmill House, Portmill Lane, Hitchin, Herts SG5 1DJ

CIRCLE NO. 098

The Editorial Team of .EXE [Will] wishes to make

A CALL FOR PAPERS

[A request for articles]

Forthcoming Editorial Thematic Issues:

[Topics coming up]

October - Mostly UNIX
November - GUI's
December - Back to the Future

Other ideas always welcome

[ie the lazy s*d can't think of them himself]

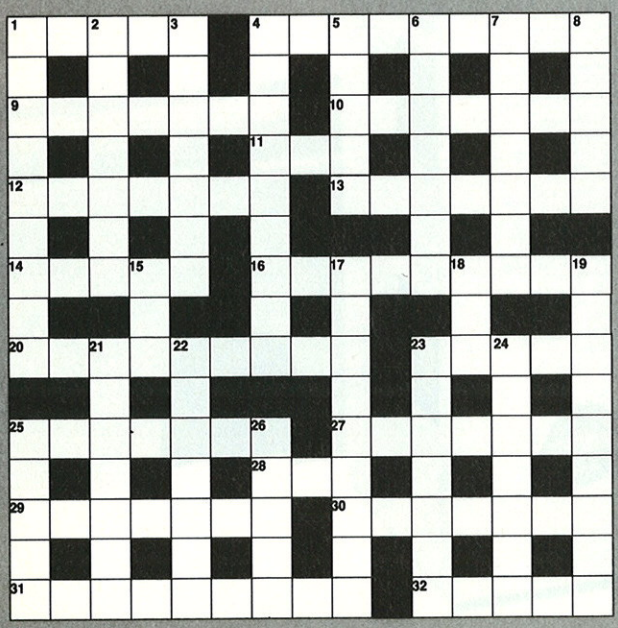
All published work generously rewarded [Ha!]

Send away for a stylish, individually bound copy of our Contributors' Notes

[A few xeroxed sheets inexpertly stapled together]

Contact The Editor, .EXE Magazine, 10 Barley Mow Passage, London W4 4PH

AUGUST .EXEWORD



ACROSS

1 Provide a running check on the first track event (5)
4 Identify the characteristic... (9)
9 ... to the goose: bend over, it's George's (7)
10 Parboil sort of transistor (7)
11 Wise bird to produce *guide* (3)
12 Hidden time to lay round ten hundred (7)
13 Sweets in European dens... (7)
14 ... of harmonic water flow ... (5)
16 ... in peace with infinite snr (9)
20 26 sets of stolen goods? (9)
23 Use silver about the orient as labels and data types must (5)
25 Extension socket or adc (7)
27 Employer action leads to fatal crash (7)
28 Palendromic language from a countess (3)
29 Person (not young) was in charge (7)
30 It follows you in France with a charge for guidance (7)
31 Madly happy when heated (9)
32 Keen sign of hesitation about some time (5)

DOWN

1 Hollerithian hardware ... (9)
2 ... stopped running right at the start? (7)
3 Global constant time-wise (7)
4 Pain in the neck on any cane strangely (9)
5 It may hold labels when output by 1dn (5)
6 I am brief digital unit with brief effect (7)
7 Madly rile guy with nasty temper (4,3)
8 Gets financial input from broken snare (5)
15 Cut part of a lax executive program (3)
17 How the wire's sheath acts (9)
18 Lump of wood holds run-time record (3)
19 Bribe that coats the pill? (9)
21 Data path tunnelled under (7)
22 Wholly unreal value (7)
23 In the bow I have long term data store (7)
24 In support digging at the bottom of the tree (7)
25 Bearing weapons in a disturbed dream (5)
26 Telecomms system with 57 degrees (2 in binary) (5)

'.EXEWORD' compiled by Eric Deeson

V	I	R	T	U	A	L	■	C	O	R	R	U	P	T
A	■	E	■	N	■	E	■	L	■	H	■	S	■	O
L	O	C	A	L	■	S	T	A	T	E	M	E	N	T
I	■	U	■	I	■	S	■	S	■	O	■	R	■	A
D	U	R	A	T	I	O	N	S	■	S	I	S	A	L
L	■	S	■	■	■	N	■	I	■	T	■	■	■	L
Y	O	I	C	K	S	■	I	F	F	A	M	I	L	Y
■	■	V	■	E	■	I	■	Y	■	T	■	M	■	■
Q	U	E	R	Y	I	N	G	■	I	S	O	P	O	D
U	■	■	■	B	■	S	■	S	■	■	■	L	■	I
A	U	D	I	O	■	■	I	N	C	R	E	M	E	N	T
N	■	A	■	A	■	G	■	O	■	P	■	M	■	H
T	U	T	O	R	S	H	I	P	■	S	I	E	V	E
U	■	U	■	D	■	T	■	E	■	O	■	N	■	R
M	I	M	O	S	A	S	■	S	O	N	A	T	A	S

JULY .EXEWORD

City **£ Excellent**
Developing substantial delivery systems for brokerage and dealing rooms, based on **OS/2, Presentation Manager** and **Token-ring LAN's**. Our client requires **OS/2 PM programming, OS/2 kernel programming** and **Microsoft C**. Any of the following are useful: **MS-Windows, NetBIOS, X.25, SDLC, Async, MS-DOS, PVCS, UNIX, Stratus VOS, Oracle, Sybase** or **Ingres**.

Wilts **to £24k**
Leading Software developers seeking additional staff with **C/UNIX** skills combined with one of: **X-Windows, MS-Windows** or **C++**.

Holland **£ neg**
Leading Chemicals organisation needs two individuals in their Process Control department:
a) Software Engineer with **C/UNIX** skills.
b) **UNIX** and **VMS** System Support.

E.Anglia **to £25k**
Major R&D development program requires at least 2:1 graduates with experience in some of the following: **C, UNIX, Real-Time, Neural Networks, C++, Voice Recognition, Speech Synthesis, OOD, DSP, OSI/X.25**.

Surrey **£17k - £21k**
Premier developer of graphical databases on Sun and Applemac is seeking two individuals: one with good **UNIX/C**, the other with **Applemac** experience. Any **IBM PC** or **Postscript**.

Hants **to £20k**
Leader of an **ISDN** Esprit 11 project consortium is seeking a software engineer with experience of **X.25** development under **C** and **UNIX**. Benefits include 10% non-contributory pension.

Herts **to £23k**
3D graphics tool developer for various workstations is seeking **UNIX** and **C** skills with one of the following: **C++, Fortran** or **CAD**. Excellent promotion prospects

Herts **£ Good**
Significant multi-media developer in the area of video conferencing and financial market information is seeking software engineers with good Graphics experience ideally with **C/UNIX** or **Assembler** with **TMS 34020** experience.

MANY OTHER OPPORTUNITIES TOO NUMEROUS TO MENTION - SO FOR FURTHER DETAILS CONTACT ANTONY BRIDGE EITHER BY PHONE OR IN WRITING.

081 876-0102 / 081 392-1514
ACUMEN SEARCH & SELECTION INTERNATIONAL
London House, 42, Upper Richmond Road West,
London SW14 8DD Fax: 081 392-1518
Out of hours telephone: 081 780-0637

ACUMEN
SEARCH & SELECTION INTERNATIONAL

SWAN RECRUITMENT

C Programmers
Swindon/Peterborough **£14-20K**
We have a variety of positions with applications ranging from commercial to datacommunications/ modems, networks.

Apple Mac
Surrey **Excellent package**
Our client has an urgent requirement for both contract and permanent staff with experience of 4D, Hypercard or MPW. Opportunities for career progression are excellent in this rapidly expanding organisation.

SAP Consultants
Europe/N America **Excellent Package**
Urgent contract requirement. Financial applications.

We are always interested in hearing from persons with skills in:
Relational Databases, 4GLs, C, Ada, UNIX,
IBM PC, Apple Mac, Dec VAX, Mainframes.
Positions located in UK and Europe. Contract and Permanent.

In the first instance call or write to:

Swan Recruitment, Bath Road Trading Est, Bath Road, Stroud, Glos GL5 3QF

Tel: 0453 755437
Fax: 0453 755436

Low Level Software Engineer **Surrey** **to £19k**
Low level skills? I should hope so, we have had absolutely no calibre BIOS and low level individuals. This leading hardware and solutions house is desperate for the right person, as indeed are we. We feel confident that the developments which will be open to you offer interest and even a little excitement! Please send your CV.

Programmer/Develop
Cambs/W London (2 positions) **£16-£20k neg**
We are a still intent on finding skilled C/UNIX and X-Windows programmers who want to move into a recognised Software House. At least 1-2 years experience is essential, with additional expertise in PCs or workstations development an advantage. Lots of development, new products and a lucrative career path are all offered in return.

Software Installation **Berks** **£20-£25k**
Not really a software job, but we ideally want a graduate with a Technical Support background, primarily in MVS, but different platforms are an advantage. Thus position is a combination of software installation, technical support and capacity planning, with a very large Multi-National Corporation.

Ingres/Oracle/Sybase **London** **£20k neg on exp**
One, two or even all of the above? Two or three years experience? We are looking for people with a good background, preferably from a major software house. You will have had experience in large projects and SSADM. There are a number of positions we can offer you, all within established consultancies / system houses.

Oracle/Technical Services **London** **£20-£25k (+car)**
Two positions requiring good Oracle skills. The first is for a Database Administrator with a large manufacturer, the second is a more general Technical Support/Consultancy role, involving - systems development, installations, on-site advising, in-house administration, etc etc. The desired platforms for this experience are VMS/PCs/Networks (in that order!)

Windows/C/Securities knowledge
London **£20k neg on exp**
Either PC or UNIX platform knowledge, and some database experience would be an advantage for this C/Windows development role. The essential requirement is for experience in the securities field, to work on in-house systems for this securities house.

PC/Network Support **London** **£16k**
Energetic, friendly support professionals needed to do exciting work running the PC network at this large enduser site. In addition to all the hardware and software support, the responsibilities will include further development and expansion of the existing systems and more installations. Interested...?

Junior Coder **Dorking** **£13.5 up**
C - UNIX - Windows - Oracle, some Mac Interfacing would also be of interest. An ideal position for a recent graduate with just a little commercial experience.

Junior Coder **Berkhamsted** **£13.5k up**
C - Windows - 4GL & SQL types of skills, one year experience upwards, rewarding software project....

Software Developer **London** **£19k**
C - Assembler - LAN - Real time and VGA card skills? This project is a good one, with excellent team structure and competent management. Good hardware expertise and structured techniques would be a bonus.

RONICOM
RECRUITMENT

5-7 Sedley Place (off Oxford Street), London W1R 1HH
Tel: 071 491 3640 Fax: 071 499 2546

CASE SALES SUPPORT ENGINEER

Prosa Software is an outstanding European manufacturer for state-of-the-art Case environments. Prosa Software is headquartered in Finland and has presence in all European countries. The tools of our Prosa Case family are used by major corporates, software development houses and computer manufacturers in the UK and Europe. Prosa tools run on all industry standard platforms including Windows 3, OS/2 Presentation Manager, UNIX, Xwindows, DEC, Sun and HP/Apollo.

We are seeking a Case sales support engineer for our UK office in High Wycombe. As a member of highly innovative and forward looking staff and fast growing company, an independent, enterprising and responsible person has an excellent opportunity to build his own career within the company.

If you are a sales-minded person, 25-30 years old and can demonstrate specific skills in the following areas

- **Knowledge of Yourdon's SA/SD/RT methodology**
- **C programming or other high-level language**
- **Knowledge of PC-DOS and Windows 3, UNIX**
- **Good organizing ability and communication skills**

then please send your application with Curriculum Vitae and other relevant documentation, quoting your salary requirements to the address below.

Prosa Software,
Mikko Tervonen - Managing Director
Tudor House, 649 London Road
High Wycombe, Bucks HP11 1EZ

prosa

CAREER OPPORTUNITIES FROM CONCURRENT

Software Engineer **£20,000** **Thames Valley**
Low level graphics software development! Multiple UNIX platforms! Build an X-Windows GUI! *Ref - SS*

Database Analyst **£24,000** **Herts**
Major database software vendor! Provide design consultancy to large multi-database corporates! Oracle or Ingres experience preferred! *Ref - DC*

VAX Technical Analyst **£22,500** **Swindon**
Pan-European support of UK's largest All-In-One network! VMS Systems Management background preferred! *Ref - TR*

Analyst or Analyst/Programmer **£21,500** **Cambs**
Water industry or other utilities background essential! Train on UNIX system! Relational Database and CASE experience helpful! *Ref - AW*

To apply for one of the above positions please telephone Alan Carnell on 0582 712976 or write with a full CV to the Harpenden address, quoting the appropriate Reference.

Concurrent Appointments
Software Recruitment

27 FIELD CLOSE
HARPENDEN
HERTS
Tel: 0582 712976
Fax: 0582 764858

Software Care

C, DOS/UNIX
Database and GUI work
Integrated Hospital Systems

Software Care, a division of CHC (UK) Limited, develops information management and knowledge-based tools for the Nursing profession.

We are looking for programmers and analysts to join our small, highly motivated technical team. Having just completed a major development project, we are continuing development in multi-user database access, rule-based artificial intelligence, and communications.

You will have a flair for imaginative design of easy-to-use programs. While we will concentrate more on your ability to make our system work easily for busy nurses rather than your particular history, you will have a good grounding in C programming and some familiarity with CASE tools and structured design techniques.

For more information, send a brief CV and covering letter (in confidence) to Samantha Hilton-Johnson, Software Care Limited, Four Church Studios, Camden Park Road, London NW1 9AY.

CHC

Software Care Nursing Systems
Member of the CHC Group of Companies

OS/2 Presentation Manager

West London **to £32K**

Unique opportunities with this dynamic software services company exploring leading edge technologies and promoting personal skills development. The successful candidate will be educated to degree level and have 3 years experience of two of the following: OS/2, Presentation Manager, Windows, C, C++, OOD

Ref: CT

C/UNIX/Graphics/Windows

South East **to £25K**

Leading the development of computer based solutions, this young innovator is offering commercially minded software engineers the opportunity to further their career at the forefront of Graphics Technology. You will probably have 2 years+ experience of C/UNIX and possibly an appreciation of graphics, windows, VRTX, porting, RISC or 680*0, microprocessor hardware.

Ref: TM

C/C++/SQL WINDOWS

Herts/London **£13-30K**

Exciting new opportunities have been created by this successful software tools specialist, who are seeking to build a development team. You will be working with C/C++, OOD, GUI, UNIX, Relational Databases and SQL; all levels of experience will be considered. Call now for further information.

Ref: CT

C./MS-DOS

Yorkshire **Circa £16K**

Do you enjoy a challenge and can you respond positively to a stimulating and varied environment? Here is a chance to make real use of your existing skills as well as acquiring new ones. Ideally you will have a minimum of 18 months C in an MS-DOS/UNIX environment with a knowledge of SQL (preferably dBASE III or IV). This is a superb opportunity to enhance and develop your software skills on new and exciting projects.

Ref: PI

For more details

Contact Paul Innes (PI) or Teresa Maddern (TM) or Christine Trybus (CT) for details on the above and many other vacancies on

0442 231691 days
or 0442 69740 eves/wkends
(fax 0442 230063).
Alternatively write to them at:

Executive Recruitment Services, Hempstead House, Selden Hill, Hemel Hempstead HP2 4LT

SOFTWARE

REAL-TIME, 'C'/UNIX, X-WINDOWS

Hayes & Stockport · to £22,000

Part of a successful British multinational, this company is committing £10M to the development of innovative intelligent control systems aimed at world-wide markets. Key new opportunities exist in the Software Development group for high-calibre graduate engineers with at least two years' relevant experience.

Initially, you would be involved with either Embedded Software or Graphical User Interface development, but opportunities exist to broaden your skills. Your involvement will be throughout the complete software development cycle, from concept to market.

For **Embedded Software**, a background in electronics is preferred, with real-time systems development experience. You must have ability with 'C' and assembler for Motorola and Intel microprocessors. Exposure to structured design techniques, SCADA, industrial networks (eg ARCNET) or embedded kernels (eg OS-9) would be advantageous.

For **Graphical User Interface** development, fluency with 'C' and experience of using X-windows in a Unix-based RISC workstation environment is ideal, whilst familiarity with CASE tools, Open-Look or Motif would be an advantage.

The initiative is high-profile, non-military, technically exciting and very challenging. The purpose-built R & D environment is first-class, career development prospects are excellent and the remuneration package is attractive.

Please send your CV and salary details, quoting ref: 023/2/EXE, to: Michael Short, Personnel Consulting Services, Wyvern House, Station Road, Billingshurst, West Sussex RH14 9SE. Telephone: 0403 785120. Fax: 0403 784988.

Personnel Consulting Services

SEARCH SELECTION AND HUMAN RESOURCE CONSULTANTS

ASH ASSOCIATES

We specialise in the Recruitment of Software Design and Support Engineers in the South East for Real Time Applications including GRAPHICS, COMMUNICATIONS, CONTROL/ROBOTICS, SIGNAL PROCESSING & MODELLING.

We URGENTLY seek highly qualified design engineers for a W.SUSSEX based client, Salaries to **£25K**.

NEW PRODUCT DEVELOPMENT

Experienced software design engineers with hardware appreciation to design NEW Data Communications/ Management Systems.

C programmers with in depth OSI, Interpreter and Compiler development experience.

Also C programmers with Graphical Interface/Comms design experience. Microsoft Windows, RS232.

Hardware design engineer with Analog and Digital design experience gained within a communications environment.

Call James Hunt or Ron Cook NOW!
TEL: (0425) 475480 (24hrs)

ASH associates

RECRUITMENT CONSULTANTS
3 Pipers Ash, Ringwood, Hants, BH24 1UF
Tel: (0425) 475480 Fax: (0425) 480807

Are YOU looking for PC Systems Analysts, C Programmers, PC Support Analysts, Unix Gurus, S/W Development Engineers, Network Specialists, Object Oriented Systems Designers, Real Time Engineers, Database Developers, Application Support Engineers,Windows Programmers, CAD CAM Experts, Graphics Specialists.....and more?

LOOK NO FURTHER THAN THESE PAGES.
Call **Chris Haill** on **081-994-6477** to find out more about this unique advertising medium. Of course they all come with their experience in ADA, PASCAL, ORACLE, C, C++, WINDOWS, CLIPPER, MOTIF, SQL, CAD-CAM, ASSEMBLER, LANs, WANs, dBase UNIX.....

.EXE – We only provide the best!

£18-£20,000 **SURREY/HANTS**

'C' AND 4GL PROGRAMMER
Executive Information Systems (EIS)

The Company

Backed by significant and secure financial resources, this recently formed company has developed the next generation of object-oriented EIS.

The Role

- Work on leading edge developments and implementation issues in open systems, multi-media and graphical user interfaces
- Enjoy a dynamic working environment with talented individuals, and exploit your strong 'C' and UNIX programming skills.

The Candidate

You have a minimum of 3 years' experience with 'C' and UNIX, and ideally some related skills including X-Windows, MS-Windows, SQL databases, 4GL's and graphics. You recognise the potential of career development within an exciting and rapidly expanding Company, and have the initiative to exploit that potential.

Call Robert Hobbs on 0252 876520, or alternatively send him your c.v. at Caves Farm House, 33 High Street, Sandhurst, Surrey GU17 8EB. Fax: 0252 890028

INPHASE
SOFTWARE LIMITED

DEVELOPMENT SPECIALISTS

Windows & 'C' Developers

to £20,000 + Benefits

With at least 2 years commercial experience in *Windows* and *'C'* development, including at least six months *Windows 3* experience, you could find yourself working on any one of several current Windows projects. Educated up to Graduate level you will be instrumental in implementing Windows based technologies into forward thinking companies. Financial experience would be an advantage, although opportunities exist in a number of development environments including end-users, financial institutions and software houses. Performance related bonuses mean that your skills and effort are fully appreciated, and competitive salary packages are on offer.

Location: The City and Home Counties **Ref: PCEX10/1**

OS/2 Presentation Manager

£20-25,000

This City based software house specialises in providing systems for the Securities industry. They are currently undergoing a major growth phase and amongst their vacancies is the requirement for someone to join a small team developing and supporting an Information Delivery System based on a networked OS/2 environment. You will be responsible for all Presentation Manager development and support, and you should have at least two years experience of *Presentation Manager* programming, together with strong *Microsoft 'C'* skills. An MS-Windows version is planned for the future, so exposure to, or interest in, Windows v.3 development would be useful. This is a high profile role with opportunities to progress into 'Hands-on' management therefore rewards will be significant, whilst dependent on experience.

Ref: PCEX10/2

'C' Developer

£18-24,000

This young, dynamic company develop advanced software solutions for the international dealing room environment. They are now looking to recruit an analyst/Programmer educated to degree level and with 2-3 years real-time *'C'* development experience under *DOS* or *OS/2*. Preferred skills also include knowledge of communications and/or datafeeds, real-time financial information systems and relational database experience, particularly *SYBASE*. The team environment is enthusiastic and proactive and all future employees must possess good interpersonal and customer liaison skills as well as a smart appearance and enthusiastic approach.

Location: City **Ref: PCEX10/3**

Real-Time Unix & 'C'

to £25,000 + Benefits

Due to its continuing success, this leading supplier of real-time systems to financial institutions across the world is currently looking to expand its development division based in Central London. A number of positions are available for software engineers with a minimum of 2 years experience, as well as consultants with over 4 years experience. It is essential that all candidates have experience of a real-time programming environment coupled with one of the following: *Unix*, *'C'* and *X-Windows*. All suitable candidates will also have exposure to the complete product lifecycle. Although not essential, any knowledge of the following would be advantageous: trading room systems, *AIX*, *Ultrix* and *VMS*. The company offers excellent career development, the opportunity for work overseas and a range of benefits on top of a competitive salary.

Location: London **Ref: PCEX10/4**

For further details of these and other permanent vacancies please contact Conrad Hills, quoting the relevant reference, on 071-734 4010 (office hours) or 081-542 8724 (evenings/weekends). Alternatively, write to McGregor Boyall, Lyndale House, 49-50 Great Marlborough Street, London W1V 1DB or fax your CV on 071-734 1297.

mcgregor boyall

IT HUMAN RESOURCING

ADVERTISERS INDEX

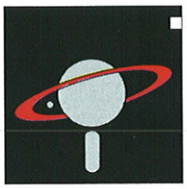
ADVERTISER	PRODUCT/SERVICE	CIRCLE	PAGE
AET	Coherent UNIX	074	61
AI International	Quintus Prolog	053	23
Applied Logic Computing	Object Oriented Programming	079	65
Bantam	Book publishers	095	86
Bits Per Second I	Graphics for Visual Basic	047	12
Bits Per Second II	Graphics Server Software	057	31
Blink Inc	Blinker fast linker	055	26
Borland I	C++	042	5
Borland II	Turbo Pascal	052	20
Brent Communications	MAX copy PROtection	097	87
BSO Tasking	Real time executive	072	59
Camel	Plotter Utilities	086	75
CEBRA Communications	Multi VGA Adapters	096	87
Clearsoft	Software Protection	099	OBC
Cocking & Drury	Smalltalk V	066	48
CTL	Copy Protection Hardware	090	81
Deepak Sareen	Project Management	064	44
DES	Software Protection	084	74
European Informix User Group	User Group	076	63
Glockenspiel	C++ Software	054	25
Grey Matter	Programming Tools	041	3
Hitex	Embedded systems debuggers	102	57
Instrumatic	C++	069	53
JPI	C++ Compilers	048	13
Kedwell	Database development	100	53
Korala	Directory Utility	091	81
Linx I	All UNIX Systems	081	67
Linx II	All UNIX Systems	093	83
Magnifeye	Software Protection Device	060	35
NAG	FORTRAN 90	056	29
Nantucket	Clipper 5.01	043	7
NEOW	Actor	061	38
Nu-Mega	Debugging Tools	044	8
PC Books	Bookshop	078	65
QA I	C Training	040	IFC
QA II	Software Testing	070	55
QBS	Clipper Add-ons	073	59
Real Time Software	CASE for Windows	059	32
Recital	RDBMS/4GL for VAX & UNIX	062	41
Ryan McFarland	COBOL compiler	045	11
S/ware Construction Company	Development Tools	083	73
Salford Software Marketing	FORTRAN for DOS & UNIX	058	31
SCL	Communications Boards	046	12
Sequiter	C Library	088	77
Smart	Embedded systems	075	63
Softlok International	Piracy Protection	082	70
Solution Systems	Programming Editor	063	43
System C	Application Generator	089	78
System Science I	Special Offer	077	63
System Science II/Lahey	FORTRAN Compilers	087	75
System Science III	RM COBOL 85 Compiler	092	81
System Star	dbms Four C	049	13
The Data Business	Software Protection	085	75
The Minix Centre	UNIX	101	57
Unipalm	Training	050	17
USA Software	Programming Tools	065	47
User Friendly	Software Copy Protection	067	50
Vleermuis	GUI tools	051	19
Zortech I	Multi Plat C++ Comps	068	51
Zortech II	C++ Video Tutorial	094	85

STOB - Barbara Cartland is a Compaq

'Scirocco, the software system from Hectare Ltd, is churning out a new 300-word love story every two seconds... The program can also be used for interpreting annual reports and accounts, and generating parliamentary replies' - The Independent.

```
C:\>scirocco /love
SCIROCCO LOADING, PLEASE WAIT...
The phone rang loudly, breaking into her thoughts. Would it be John? Would he still want to go to Corfu with her? Despite herself, her hands trembled as she picked up the phone.
'Yes?'
'Is that you, darling?' His dear voice boomed in her ear.
'Oh yes, darling!' Her hands were trembling on the receiver.
'Did you have a good day?' boomed out his dear voice.
'Not bad. How was yours?' she replied, trembling.
'It wasn't bad ^C
', he boo ^C
*USER BREAK*
C:\>scirocco /finance
SCIROCCO LOADING, PLEASE WAIT...
The phone rang loudly, breaking into her thoughts. Would it be John?
'Yes?'
'Hello Jane. The report has finally come through.'
'Well?'
'They're going in. Twenty-seven million fully paid up, retrospective option voting shares.'
'Twenty-seven million? But what about the tax liability?' She trembled at the thought.
'Tax liability?' he boomed. 'Tax lia ^C
*USER BREAK*
C:\>scirocco /parliament
SCIROCCO LOADING, PLEASE WAIT...
The phone rang loudly, breaking into her thoughts. Would it be John?
'Is that you John?'
'Before I answer that question, I would like to state categorically, emphatically and clearly that the question of identity is...'
She interrupted him, trembling.
'But John, it's me!'
'No, no; you've had your chance to speak, now it's my tu ^C
you in the so-called press ^C
the right honourable member pressing against her ^C ^C ^C
*USER BREAK*
C:\>scirocco /love
SCIROCCO LOADING, PLEASE WAIT...
'Oh BB', she sighed, nuzzling his shoulder, 'you're so much older and richer than me. I know our love will never fail. Why don't we get together and make an operating system?'
'Yes. Yes! We must do it!', he breathed. 'And we must call it "OS/2".'
'OS/2?' The elegant line of her brow wrinkled. 'OS/2? I thought a far better name would be Win ^C
*USER BREAK*
*WOT?*
*Are you SURE you don't want to know how it finishes? (Y/N)
```

.EXE



PLANET
INTERNATIONAL

Planet International was launched as the recruitment wing of a leading PC training company specialising in database and programming products - Delta Dimensions Ltd. Delta as an Authorised Training Centre for many leading software houses, such as Nantucket, Fox Software International, Ashton-Tate and Lotus, enables Planet International to make full use of the considerable technical expertise at hand, to place quality candidates in appropriate positions.

Currently our clients are urgently seeking the following personnel:-

Senior MIS Analyst

C £17,000 + Mort. Sub

Graduate required (pref. degree in computing) with at least 1 year's commercial experience in "C", FoxBase and Clipper ideally within the Financial Services Sector. Must have good oral and written skills. Some European travel required.

Consultants

£NEG.

Exp. in VAX, UNIX and Sybase database design and implementation. Must be capable of working in a consultancy role that is very much hands-on. Good oral and written skills are essential. Other consultancy positions are also available.

Analyst/Programmer

to £30,000 NEG.

The successful candidate must be fully conversant in "C" and Clipper (Summer '87/5.0), & FoxPro or dBase within the Banking Sector. Experience of Fixed Income Products, Eurobond & Gilts. Good promotional prospects.

Clipper Trainer

C £16,000

Highly presentable person required to train in Clipper. Must be fully conversant with Clipper Summer '87 and 5.0. Knowledge of various database languages would be a distinct advantage. Must be able to train groups of up to 10 delegates.

Programmer/designers

£NEG.

Experience in one of the following combinations is essential: Unix, "C", Oracle, Ingres. Microsoft Windows 3 and/or X11 Windows. All benefits associated with a major company.

Systems Analysts/ Senior Systems Analyst

£ to 30,000

Graduate (degree in a numerate discipline) with experience in PC Lan, SQL, Paradox, "C", Windows preferably gained in either a financial, operations or scientific (engineering/ exploration) environment. Also required Analyst Programmers with CASE skills or experience on HP3000 or IBM MVS. Must have good communication skills. All the benefits associated with a major company.

We are always looking for experienced Clipper, "C" and FoxPro personnel for these and other current vacancies call:

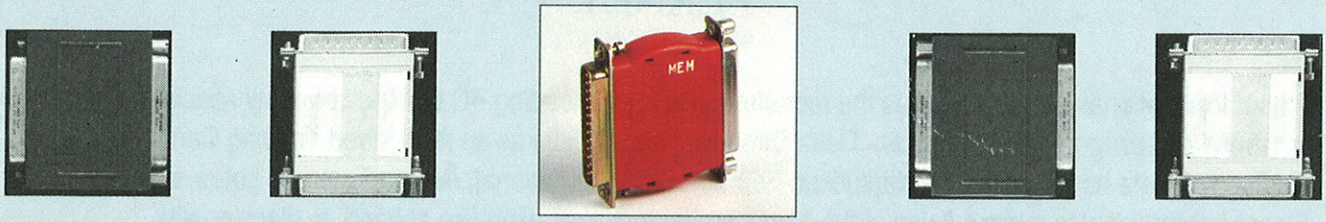
Carol Groves or Robert Ray on 071-404-8815
or
Fax your CV to us on 071-242 9901

Please direct any training enquiries to Georgette Rowland on 071-831-6991

Planet International. Consultancy & Recruitment. A division of Delta Dimensions Ltd.
20-21 Tooks Court, London EC4A 1LB

SOFTWARE PROTECTION!

Can You Spot The Difference?

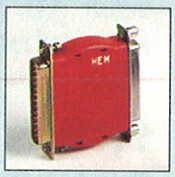


- ☑ High-level security keys
- ☑ Assembler-based, customer specific, encrypted interrogation routines
- ☑ Over 70 languages supported
- ☑ Compatibility - due to 8 years experience & 600,000+ keys sold
- ☑ Reliable on-going support
- ☑ Physically unique keys for each customer
- ☑ MS-DOS, OS/2, UNIX, XENIX, WINDOWS 2 & 3, "MACINTOSH"
- ☑ Transparent operation with most peripherals on market
- ☑ Free language updates
- ☑ Parallel, Serial, Mac ports

OVER 3,100 CUSTOMERS WORLDWIDE HAVE!

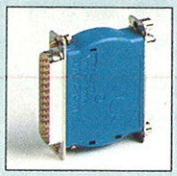
Electronic Key

The ideal device for identically produced software packages. Uniquely wired with customer code and a software code. Uses Assembler based program, decryption interface and random values.



1 Word Memory Key

Custom hardware wiring allows the developer total control over information stored in the key. 2 bytes of memory allows several packages to be protected with just the one key.



31 Word Memory Key

For multiple software protection schemes. 31 words of 16 bits of non-volatile dynamically programmable memory. Its capacity to store information provides virtually limitless power. Flexible protection scheme can be modified on-site during operation of software package.

Micro Processor Key

Provides the ultimate in software security. Not tied to any language or O/S. 8 bit microprocessor powers from RS-232 level. Requires no power supply. For PC terminals, minis, & others using RS 232 C comms. Used on workstations. This key is effectively a computer.

Macintosh Key

Extremely powerful & customised protection for the Mac. 31 words available for random storage. Providing unequalled protection the Macintosh Memory Key connects to the SCSI 25 pin port and operates transparently.

MICROPHAR... The European Standard!

For a no-nonsense informative discussion on how our dongles can better protect your profits, please call us on:

091-378 9191

CL

CLEARSOF
Langley Moo
Telephone: (

Please send me full details on your protection systems.

Name: .....

Position: .....

Address: .....

.....

.....

.....

.....

.....

.EXE 7/91

.EXE
Please mark 'Later' if appropriate
Notify 050 SAH for address change
~~EC DISPLAY 2 WKS~~ ~~LIBRARY, EC~~ Date Seen 19.8.91
~~PJ WINFIELD~~ ~~INF/TECH, EC~~
~~DG MARK~~ INF/TECH, EC
EC LIBRARY EC

02686872/V6*N3/EC.S/1